



Road and Building Reconstruction from 3D LiDAR Point Clouds: A Scoping Review

N. Krishnakumar¹ · Jaya Sreevalsan-Nair¹ · Prashant Goswami²

Received: 30 January 2026 / Revised: 27 March 2026 / Accepted: 31 March 2026
© The Author(s) 2026

Abstract

Recent advancements in semantic 3D city building reconstruction have achieved high levels of geometric detail, enabling a wide range of applications in urban planning, digital twins, and smart city systems. However, the state of the art (SOTA) has three glaring gaps. Firstly, building and road reconstruction are often treated as decoupled tasks, overlooking the mutual benefits of integrating both within a unified modeling framework. The integration is beneficial, as roads could serve as reliable spatial priors that aid in the determination of the optimal layout of buildings. Secondly, despite the availability of large-scale point clouds from airborne and mobile LiDAR (Light Detection and Ranging) systems, most existing workflows remain manual or semi-automated, relying on rule-based modeling and expert intervention. Finally, there is a lack of open datasets for an integrated 3D building-road reconstruction. A scoping review of the existing work is much needed to identify these gaps. This comprehensive unifying review of existing techniques includes both building and road reconstruction from point clouds, spanning both parametric and data-driven approaches. Further, a few key building and road reconstruction methods are implemented on an example open dataset to identify challenges. This pipeline treats road modeling as a prior; thus, it is implemented before the reconstruction of buildings. This example highlights key challenges, such as pose correction, ambiguity in footprint estimation from MLS data, and semantic inconsistencies in the reconstructed meshes, that need to be studied further.

1 Introduction

Semantic 3D city models have become essential components in modern urban analytics, digital twins, and smart infrastructure systems. Unlike conventional 3D representations, these models embed semantic information and structure, allowing for advanced querying, analysis, and simulation. They provide a holistic representation of diverse

urban entities, such as buildings, roads, vegetation, and utilities within a coherent spatial hierarchy [1, 2]. Among the many elements within a semantic city model, buildings and roads serve as foundational elements providing a sense of “permanent structures.” They are significant as they give the structure and physical layout of the city. As a result, they serve as reliable spatial anchors and base layers for relatively long-term modeling tasks. A city model that minimally captures both roads and buildings is both structurally complete and semantically robust, supporting a wider range of downstream applications. While such a model can be manually curated, its complexity calls for automation in reconstructing the geometry of the model from multi-source datasets for improving adoption.

In practice, standards for 3D city models, such as CityGML [3–5] (versions 1–3) and CityJSON [6], formalize the digitization and software development of such structures, defining not only object types but also different levels of detail (LODs) tailored for varying applications, from visualization to energy simulation and disaster preparedness. These standards are widely adopted due to their reliability, interoperability, and support for scalable

✉ Jaya Sreevalsan-Nair
jnair@iiitb.ac.in

✉ Prashant Goswami
prashant.goswami@bth.se

N. Krishnakumar
krishnakumar.n@iiitb.ac.in

¹ Graphics-Visualization-Computing Lab (GVCL), International Institute of Information Technology Bangalore (IIIT-B), 26/C Electronics City, Hosur Road, Bangalore, Karnataka 560100, India

² Department of Computer Science (DIDA), Blekinge Institute of Technology (BTH), Valhallavägen 10, Karlskrona 371 79, Sweden

data exchange. While CityGML provides a comprehensive, XML-based schema for semantic representation, CityJSON caters to a subset of the CityGML standard, offering a developer-friendly, lightweight alternative optimized for web applications. Through collaborations between academia, industry, and government organizations, several web-based platforms now enable interactive access and visualization of reconstructed cities, facilitating efficient exploration [7–9].

Multi-source datasets are needed for programmatically generating such models, which include 2D images and 3D point clouds of projections and scans of the region, respectively [10]. The choice of datasets used for the task depends on the task it accomplishes. Given 3D point clouds directly maps to the *spatialization* of the region of interest, the information retrieval across *registered* datasets should maximize that of the point clouds. An effective reconstruction workflow must also carefully consider the order and interdependence of urban elements [1, 2, 11].

Example 1 An illustrative example of the reconstruction stages, such as raw point clouds to segmented data, followed by LOD1 and LOD2 equivalent models for both roads and

buildings, is shown in Fig. 1, providing a visualization of the outcome of the desired reconstruction pipeline.

Crucially, road reconstruction must precede building reconstruction in an effective modeling workflow. Roads act as spatial priors, providing cues for pose estimation, axis alignment, and orientation normalization, and become extremely valuable when dealing with city modeling (Fig. 2). Accurate road models also support the inference of spatial relationships and help resolve ambiguities caused by occlusions or inconsistent point densities. Starting with roads thus improves both the quality and reliability of the subsequent building reconstructions [13]. However, the road and building reconstructions are treated separately and decoupled in the state of the art.

Despite significant advances in point cloud acquisition technologies, most practical city modeling pipelines remain manual or semi-automated. Parametric modeling techniques such as 3Dfier [14] and City3D [15] are widely used due to their robustness but require expert intervention and are not generalizable, especially across different structural layouts and architectural styles. On the other end, data-driven approaches promise automation but often struggle with large data, the presence of noise, etc. This motivates the need for scalable, automated, and modular workflows that

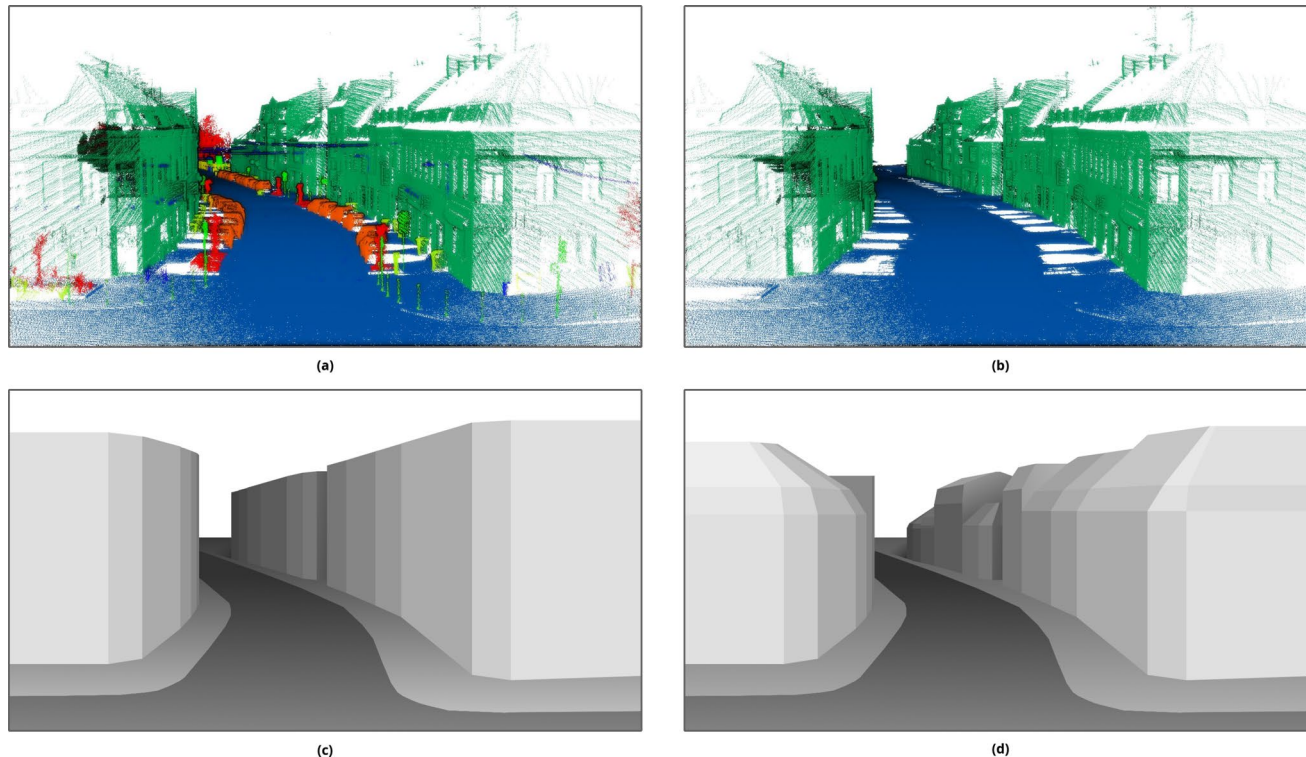


Fig. 1 Illustration of segmented point clouds and the reconstructed scene: (a) segmented point clouds, (b) filtered road and building points, (c) LOD1-equivalent reconstruction, and (d) LOD2-equivalent

reconstruction of the NPM3D/Paris-Lille dataset (Lille-2 tile) [12] (author's own source)

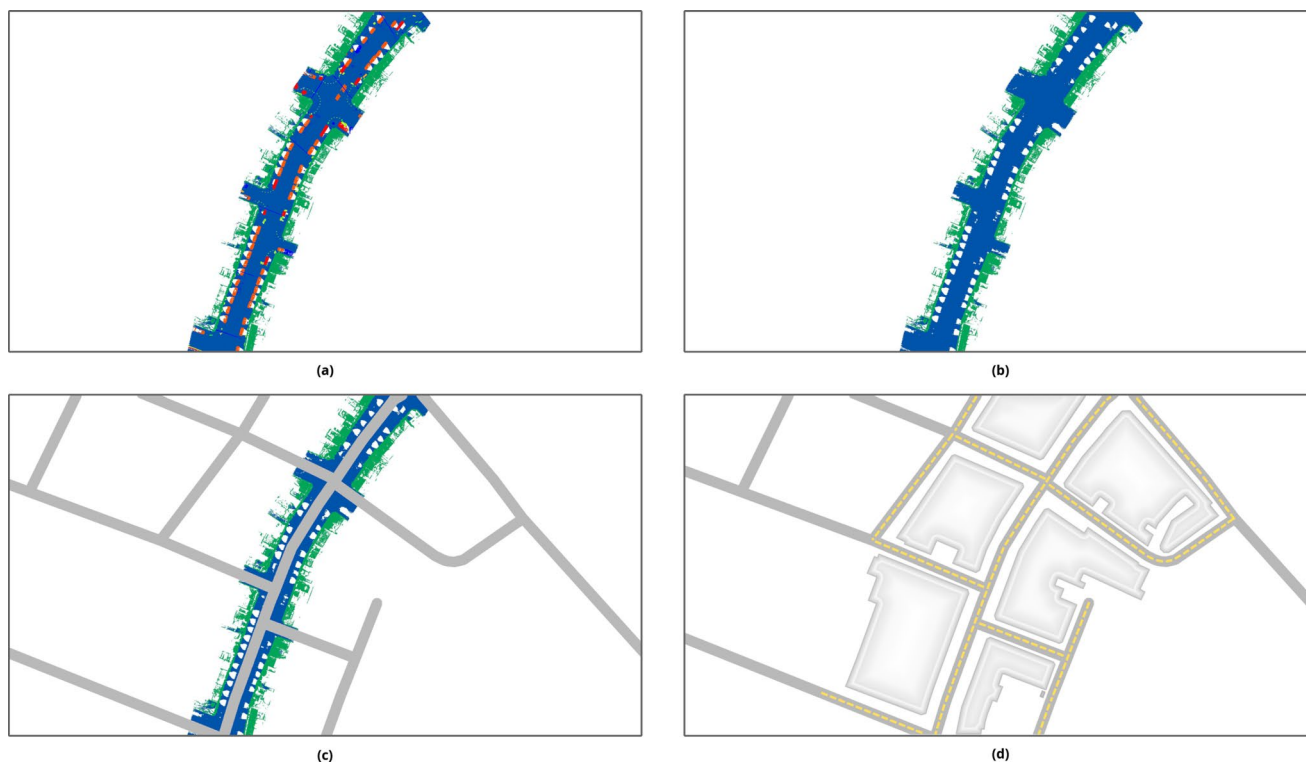


Fig. 2 Illustration of segmented point clouds and the reconstructed scene of the NPM3D/Paris-Lille dataset (Lille-2 tile) [12]: (a) segmented point clouds, (b) filtered road and building points, (c) recon-

structed roads compared with the filtered road and building points, and (d) LOD1-equivalent reconstruction with the road as a structural prior (road centerlines are highlighted in yellow) (author's own source)

can integrate parametric robustness with data-driven flexibility and adapt to a variety of urban data contexts.

LiDAR (Light Detection and Ranging) is an *active* sensing modality for large-scale 3D urban mapping, offering accurate geometric information independent of illumination or texture. It is an active sensor because it emits its own light beams for laser scanning (LS). Compared to vision-based or multimodal approaches, this work focuses on LiDAR data as the central input for reconstruction, owing to its inherent 3D spatial characteristics. Functionally, *pulse* LiDAR captures geometry by emitting laser pulses toward surfaces and recording the time of their return. Each pulse reflection corresponds to one or more 3D points, and together these points form a detailed spatial representation, i.e., a *3D point cloud*. A 3D point cloud is a set of 3D position vectors of an unordered list of *point objects*, each typically defined by its coordinates (x, y, z) , and optionally, additional scalar fields, such as color, intensity, or reflectance. In contrast, *waveform* LiDAR is a pulsed variant that also records the continuous waveform data for each pulse. The focus of this scoping review is on the LiDAR point clouds in the *photogrammetry* and *remote sensing* domains.

The domain of digital reconstruction suffers from *data scarcity*, as most available datasets either lack high-quality 3D mesh ground truths or offer only partial scene

descriptions or scans. While generic datasets like ShapeNet [16] or ModelNet40 [17] provide widely used point clouds and meshes, they are limited to individual objects and do not represent a scene of an urban environment with multiple objects. Moreover, real-world point clouds often suffer from noise, occlusions, differential sampling of objects, and non-uniform densities. For instance, in ALS (Aerial Laser/LiDAR Scanning), pedestrians have much fewer samples than a building with a large footprint, and in MLS (Mobile Laser/LiDAR Scanning) scans, the vertical façades are over-sampled, while omitting rooftops. Additionally, auxiliary metadata such as GPS traces, road centerlines, scanner/camera orientation, or multimodal sources are often essential for *complete* modeling. However, they are not always available owing to the high costs of data collection in large regions, such as cities.

For the geometric modeling needed for 3D reconstruction, generic mesh generation techniques frequently fail in urban scenes. This is attributed to the geometric complexity and architectural variability in the scenes. Models trained on shape-based datasets may perform well for object instance reconstruction but often struggle to generalize to large-scale, city-level point cloud datasets. As a result, the generated meshes may contain voids, topological artifacts, or fail to semantically align with real-world structures.

Moreover, MLS-based workflows demand additional steps such as footprint extraction, increasing pipeline complexity, and introducing new points of failure.

These limitations highlight the need for a unified modeling framework that systematically connects the reconstruction of key urban components, specifically roads and buildings, within a common computational workflow. This scoping review summarizes the state of the art and presents a unified modular pipeline to address these gaps. The pipeline unifies both road and building reconstruction for a more complete 3D city modeling. This pipeline covers tasks such as road geometry extraction, centerline prediction, pose estimation, curve modeling, and surface reconstruction for roads, and subsequently performs footprint detection, façade modeling, mesh generation, and roof reconstruction for buildings. Implementing the pipeline for an example has further helped in highlighting challenges in practice, which also provides a future direction to the current study.

The contributions of this scoping review are to:

- provide an analysis of approaches in the existing methods in city modeling with road and building reconstruction from 3D LiDAR point clouds, emphasizing the need for roads as spatial priors for building reconstruction,
- compile a holistic review of data standards, methods, and widely used open datasets,
- highlight open challenges and propose future directions towards a unified modular pipeline for 3D city modeling and its automation.

2 Methodology

Several works have reviewed 3D reconstruction and semantic modeling of cities from point clouds across multiple disciplines (Table 1) in the form of reviews, surveys, or systematic literature reviews (SLR). Ma and Liu [2] provided an early comprehensive overview of reconstruction techniques in civil engineering, detailing preprocessing and modeling workflows relevant to LiDAR-based systems. Wang et al. [1] offered a broad survey of urban object reconstruction, classifying algorithms by object type, data source, and model representation, and highlighting scalability and challenges with LOD-representation. Focusing on buildings, Xu and Stilla [11] analyzed LiDAR-based acquisition, processing, and geometric modeling methods, emphasizing gaps in automation, accuracy, and interoperability. More recently, Buyukdemircioglu et al. [18] reviewed deep learning (DL) approaches combining imagery and point clouds for 3D building modeling, while Yu et al. [13] and Huang et al. [20] discussed advances and benchmarks in road and surface reconstruction, respectively. As a complementary,

Wysocki et al. [21] surveyed semantic 3D city datasets, identifying limitations in dataset standardization and model evaluation.

While prior surveys extensively cover specific reconstruction tasks or data modalities, there still remains a gap in an integrated, end-to-end workflow. This work aims to address it through a scoping review that unifies preprocessing, segmentation, modeling standards, and evaluation within a generic semantic 3D city modeling framework.

2.1 Overview of the Review Process

Rather than adhering to a rigid systematic review protocol, the scoping analytical review framework in this study combines the workflow in 3D reconstruction from point clouds for the permanent structures in a semantic city model. The emphasis here is on starting with the methodological diversity for different structures, namely, roads and buildings, to identify recurring workflow patterns, and then highlighting emerging trends in the reconstruction of these structures.

Objective and Scope: The primary objective of this review is to consolidate and analyze existing research on urban-scale 3D reconstruction, with an emphasis on buildings and roads, the two most semantically and geometrically significant components of city models. The review spans both parametric and data-driven approaches, including methods that leverage LiDAR, photogrammetry, or fused data sources.

Search Strategy: The literature used in this review was collected through a combination of targeted keyword searches and the use of curated relevant research by an expert in the domain of 3D city modeling. Scopus and different core collections of Web of Science (WOS) were selected as primary databases, and Google Scholar was used as a secondary source, supplemented by references cited in key review papers and technical reports. A subset of articles that reflect both classical geometric modeling and recent data-driven approaches was finally selected, instead of performing an exhaustive, comprehensive search. The search focused on domain-specific keywords, drawn from two categories. The lists of keywords are:

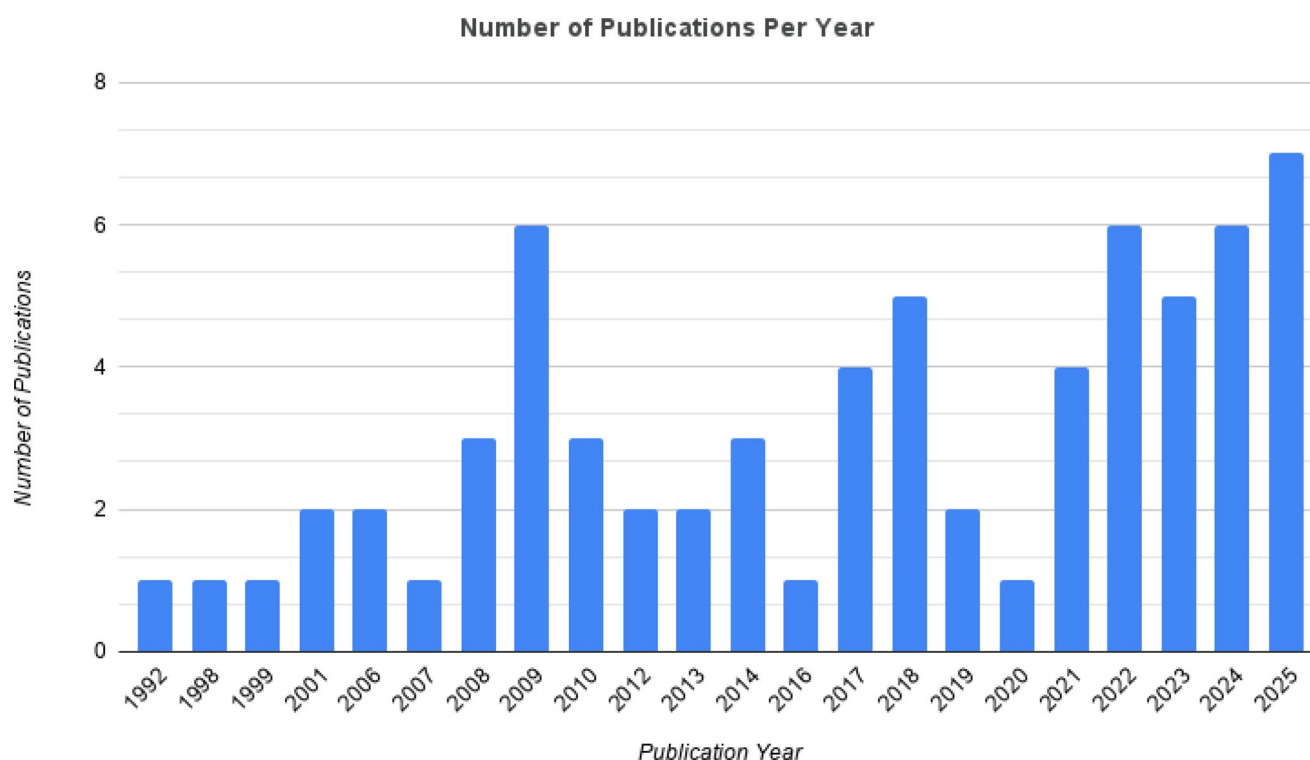
- **Road Reconstruction:** “Road modeling,” “road surface reconstruction,” “centerline extraction,” “curve fitting,” “spline modeling,” “geometric design,” “point cloud segmentation,” and “surface fitting.”
- **Building Reconstruction:** “3D city modeling,” “building reconstruction,” “CityGML,” “footprint extraction,” “parametric modeling,” “roof reconstruction,” “façade modeling,” “mesh generation,” “geometric modeling,” “Building Information Model (BIM),” “digital twins,”

Table 1 Relevant existing surveys and reviews, on semantic city 3D modeling, with the number of articles (N_A)

Reference	Year	Application Domain	Scope	SLR	N_A
Ma and Liu [2]	2018	Civil engineering (construction sites, buildings, bridges, roads) and Mechanical, Electrical, and Plumbing(MEP) systems	Reviews 3D reconstruction techniques, covering LiDAR-based processes, algorithms, applications, challenges, and future research directions	Yes	100
Wang et al. [1]	2018	3D reconstruction from point clouds, including buildings, roads, bridges, vegetation, utilities, and complex free-form structures.	Reviews and evaluates urban reconstruction algorithms, focuses on object type and input/output data, and highlights adaptability, interactivity, and great detail modeling considerations.	No	197
Xu and Stilla [11]	2021	Building and civil infrastructure reconstruction	Reviews acquisition, processing, and reconstruction techniques for buildings and infrastructure, highlighting limitations, research gaps, and future directions.	No	258
Buyukdemirciogluet al. [18]	2022	3D building reconstruction using deep learning and Earth Observation data (aerial/satellite imagery, point clouds)	Reviews deep learning and conventional methods for generating 3D building models, discusses their strengths and weaknesses, and highlights challenges and future directions for city-scale modeling.	No	71
Abreu et al. [19]	2023	Scan-to-BIM and Scan-vs-BIM applications	Procedural point cloud modelling approaches for BIM integration	No	77
Yu et al. [13]	2024	Road reconstruction and 3D modeling from LiDAR point clouds	Review of existing methods for road segmentation, feature extraction, and 3D modeling using parametric, surface, and hybrid approaches	No	143
Huang et al. [20]	2024	Surface reconstruction from point clouds	Survey and benchmark of algorithms for 3D surface reconstruction	No	150
Wysocki et al. [21]	2024	Semantic 3D city model datasets, with a focus on data availability and characteristics rather than modeling itself	Reviews global semantic 3D city model datasets, compares object level statistics, and highlights suitable datasets for testing and 3D reconstruction research	No	42
This study	2026	Semantic 3D city modeling, with focus on building and road reconstruction	Covers the full end-to-end pipeline including preprocessing, segmentation algorithms, modeling standards, evaluation metrics, and provides a generic framework, unlike existing works that focus mainly on modeling	No	155

Table 2 Inclusion-exclusion criteria used for database collection in this scoping review

Inclusion Criteria	Exclusion Criteria
<ul style="list-style-type: none"> • Papers that utilized LiDAR or multimodal datasets using LiDAR were included. • Pre-2010 milestone works foundational to preprocessing or reconstruction workflows are considered. • Data primarily from Mobile (MLS) and Airborne (ALS) LiDAR, and also, a limited number of Terrestrial (TLS) datasets and backpack LiDAR are considered. Input data typically consisted of raw, oriented, or RGB point clouds. • The output representations in the selected works were generally watertight LOD-models or meshes. For roads, methods producing polyline or surface-based representations were also considered. • Few studies focusing specifically on footprint extraction or roof modeling were included, given their crucial role in the overall reconstruction pipeline. 	<ul style="list-style-type: none"> • Papers that do not utilize LiDAR data or utilized Satellite LiDAR (SLS) were excluded from consideration. • Older digital twin papers (prior to 2008) were generally excluded, with a few exceptions made for seminal works that significantly influenced later research. • Generic surface or mesh reconstruction papers not specific to urban scenes are excluded. • LOD4 and indoor modeling studies were not included, as the review focuses on practical modeling up to LOD3. • Studies dedicated to non-building entities (e.g., bridges, tunnels) or heritage monuments are excluded, since the scope is city-scale reconstruction • Papers that belong to highly domain-specific subfields such as civil or infrastructure engineering without reconstruction relevance were excluded.

**Fig. 3** Year-wise distribution of the articles reviewed in this paper. (author's own source.)

“procedural modeling,” “MLS point clouds,” and “LiDAR.”

The curated content was directly collected from querying these keywords to give a representative coverage of the literature for a range of component-wise to city-scale models.

Inclusion and Exclusion Criteria: To ensure the review remained focused, consistent, reproducible, and relevant to the research objectives, a set of inclusion and exclusion criteria was defined (Table 2). These criteria were applied

during the curation and refinement of the collected literature, to retain only the studies aligned with LiDAR-based city-scale reconstruction methods and semantic 3D modeling, particularly of roads and buildings. The review covers research published between 2008 and 2025 (Fig. 3), with approximately 70% of the papers appearing in the period from 2013 to 2025.

Through this structured yet flexible review process, the selected body of literature provides a comprehensive view of methodological progress in LiDAR-based 3D

reconstruction. This work has reviewed articles spanning across 2.5 decades, which build the foundation to analyze, categorize, and interpret the evolution of techniques for road and building reconstruction.

3 3D Reconstruction Pipeline

The primary goal of a 3D reconstruction pipeline is to automatically or semi-automatically create structured, semantically-rich 3D models of urban environments from point clouds (Fig. 4). This raw data often includes other scalar fields such as RGB values, intensity, and scan angles, which can be leveraged alongside computed geometric features to aid reconstruction and refine the output models. Since the complexity and scale of cityscapes demand robust and

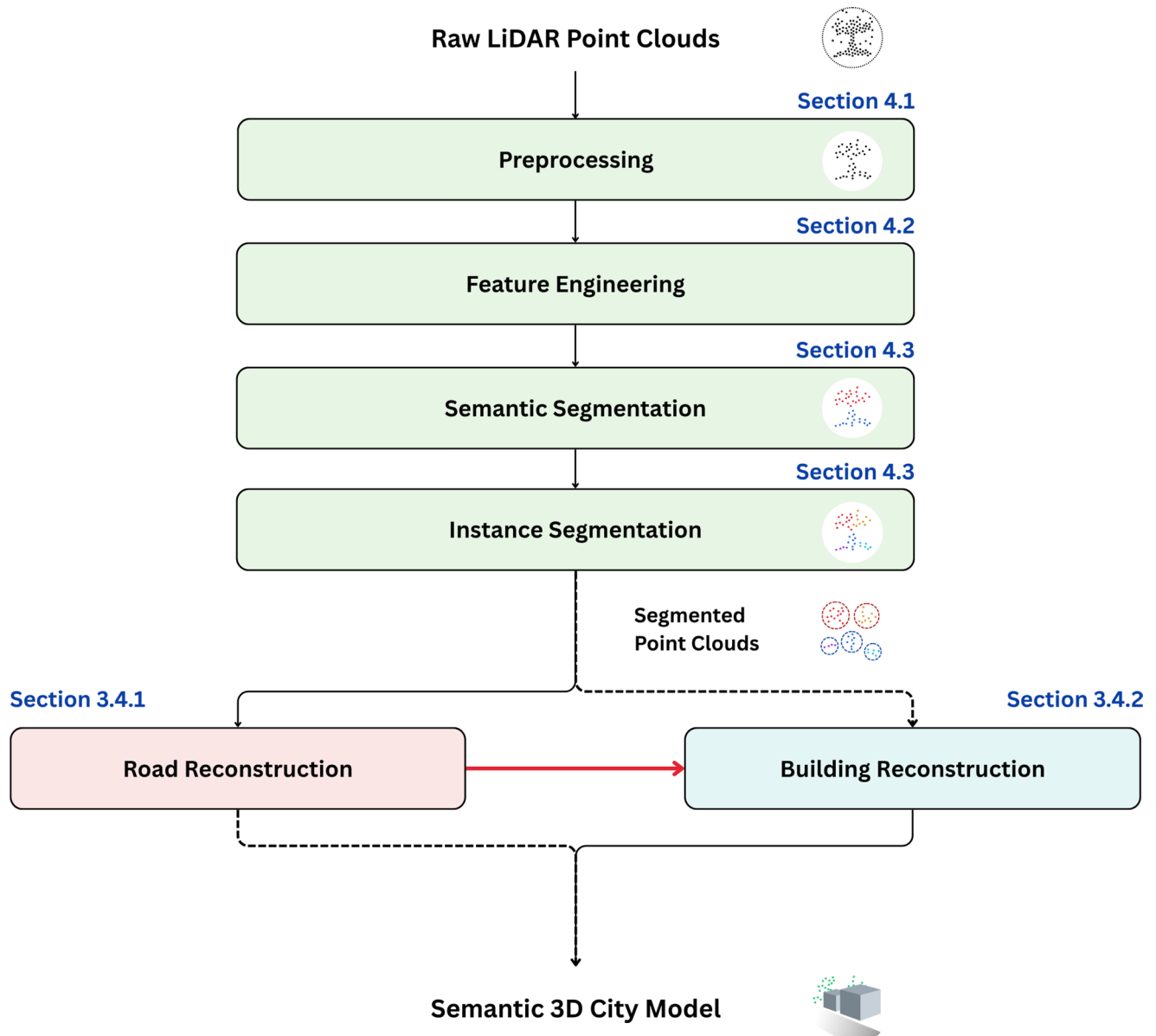


Fig. 4 Overview of the reconstruction pipeline, illustrating the processing stages from raw LiDAR point clouds through preprocessing, feature engineering, semantic and instance segmentation to downstream geometry reconstruction. Dashed lines denote the conventional approach, where both road and building reconstruction are performed independently and in parallel from segmented point clouds, as is

typical in current state-of-the-art methods. The red arrow represents a major contribution of this work, where road reconstruction outputs are used as a structural prior to inform building reconstruction, replacing the direct dashed-line path. Together, the road and building reconstruction modules form an integrated pipeline that produces a semantic 3D city model. (author’s own source)

specialized workflows, this section reviews the prominent methods and architectural patterns for this task, beginning with foundational generic surface reconstruction techniques to establish a baseline. This section will then progressively narrow the focus to specialized pipelines for road and building reconstruction, highlighting the evolution from classic geometry-based algorithms to modern data-driven approaches.

Since the *segmentation* step is required for distinguishing road and building points from other point samples, the process is briefly introduced here, and further details are given in Sect. 4.3. *Segmentation* is the process of determining the specific class label of each point in the point cloud based on its contextual characteristics. *Semantic segmentation* is a type of segmentation where the points are given object class labels to provide semantic context. *Instance segmentation* is where instance IDs are given as class labels for countable objects.

3.1 Data Standards

Standardized data models are essential for ensuring interoperability, consistency, and scalability in 3D city modeling. They define how geometric, topological, and semantic information is structured, stored, and exchanged seamlessly across diverse applications and platforms. In the context of city-scale reconstruction, buildings and roads represent the most fundamental and semantically rich elements, and their corresponding standards, such as CityGML, CityJSON, and IFC (Industry Foundation Classes), form the foundation for data integration and exchange. The IFC standard, an open and globally recognized data model published under a Creative Commons license and formalized as ISO 16739 [22], provides standardized digital descriptions of built assets within the broader BIM framework.

CityGML, developed by the Open Geospatial Consortium (OGC), is a widely accepted standard for 3D city data. It provides mechanisms for model representation, storing, and sharing of detailed urban information. The improved CityGML 3.0 [5] supports up to five Levels of Detail (LOD0 to LOD4) and now includes extended capabilities for terrain, building, and indoor modeling. Despite being XML-based and relatively verbose, CityGML remains the most comprehensive and semantically rich framework for 3D city modeling for nearly two decades, through the previous versions, namely CityGML 1.0 [3] and CityGML 2.0 [4].

CityJSON complements CityGML as a simplified, lightweight JSON encoding of the same conceptual model [6]. It is optimized for web applications and data exchange, significantly reducing file size and parsing complexity, and is also semantically consistent with CityGML.

Although not directly tied to city-scale 3D reconstruction, BIMs [23] provide detailed, object-oriented representations focused on design, construction, and lifecycle management at the building scale. The Industry Foundation Classes (IFC), developed by buildingSMART, form the backbone of open BIM data exchange. The broader IFC framework includes the Data Dictionary (bSDD), Information Delivery Manual (IDM), Model View Definition (MVD), and BIM Collaboration Format (BCF). IFC defines an open and neutral data schema for representing building components and their attributes, ensuring interoperability across architectural, engineering, and construction workflows. While BIM standards primarily operate at the micro (building) scale, CityGML functions at the macro (urban) scale. Due to that, linking IFC and CityGML enables seamless BIM–GIS integration, bridging detailed building models with broader city contexts.

In practice, however, BIM models might not be available for most existing buildings. In such cases, Tarsha Kurdi et al. [24] show that building indicators such as footprint area, volume, and intensity indices can be derived directly from airborne LiDAR point clouds, providing a viable alternative. These LiDAR-based estimates correlate well with classically derived cadastral indicators while capturing richer 3D information, such as building volume and multi-story area, which 2D representations cannot provide, reinforcing the practical value of standardized multi-LOD frameworks.

In the case of roads, the Level of Granularity (LoG) in CityGML 3.0 serves a role analogous to the Level of Detail (LOD) used for buildings. LoG defines hierarchical representations of road networks, ranging from simplified centerline-based abstractions (LoG0) to detailed surface and lane-level models (LoG1–LoG3). A detailed discussion can be found in the work of Wysocki et al. [21]. Standardization efforts such as those within the OGC framework and various national mapping initiatives, formalize these granularity levels, ensuring compatibility across applications, e.g., navigation, traffic simulation, infrastructure planning, etc.

Together, the modeling frameworks CityGML, CityJSON, and IFC-based BIMs establish a consistent foundation for semantic urban modeling, enabling interoperability between geometric reconstruction methods and high-level city information models with semantic richness and computational efficiency [25]. Most of the papers reviewed in this work focus on one or more of these frameworks, reflecting their central role in current urban reconstruction research for applications, such as visualization, simulation, and analysis for digital twins.

3.1.1 Overview of LOD-Standards

In urban modeling, the concept of levels of detail (LODs) defines the granularity and complexity of 3D building representations. An LOD specifies how much structural detail is included in a model, ensuring consistency and interoperability across different applications such as visualization, simulation, urban planning, and reconstruction. Lower LODs provide coarse but scalable representations suitable for city-wide analyses, whereas higher LODs capture increasingly detailed geometric and semantic information. According to CityGML and related standards, the LOD-structure is additive, where a higher LOD subsumes the lower LODs and adds extra finer information. The definitions, as illustrated in Fig. 5 are:

- **LOD0** : Represents 2.5D surface models or terrain
- **LOD1** : Adds block models or simple extrusions from footprints
- **LOD2** : Adds roof structures
- **LOD3** : Adds façade details (e.g., windows, doors, etc.)
- **LOD4** : Extends to interior structures

2.5D surface models define a locus of points in 3D space lying on a surface, which is geometrically 2D. Also, a 2.5D surface is equivalent to a set of *projected* points in the *xy*-plane, which carry height (*z*) information, thus being analogous to a *height map*.

3.2 Generic Surface Reconstruction Methods from 3D Point Clouds

Lorenzen and Cline [27] introduced Marching Cubes (MC), a classic algorithm that processes scalar fields in volumetric data. It computes an *isosurface*, which is a locus of points with the same scalar value. In the method, 3D space is divided into a grid of cells or cubes, and at the cube vertices, i.e., grid points, a binary value encoding the relationship between the scalar value at the vertex and the isosurface value is stored. Then, triangles are generated in each cube based on a pre-computed lookup table that corresponds to

these binarized values at the cube vertices. The lookup table provides information on how the triangulation of the reconstructed (iso)surface is to be generated. Since isosurface extraction requires a scalar field as an input, *signed distance fields* derived from point clouds [28] have been used to generate isosurfaces of different values [29, 30], also referred to as *level sets* to generate surfaces. Recently, a DL variant of the use of MC for explicit surface representation has been realized using a differentiable variant of MC [31].

The ball pivoting algorithm by Bernardini et al. [32] generated a triangle mesh, which is an orientable manifold, from oriented point clouds, by incrementally growing the surface through a region-growing triangulation process. Starting from an initial seed triangle selected based on local proximity and normal consistency, a ball of fixed radius is pivoted around active boundary edges to identify neighboring points. Triplets of points touched by the ball form new triangles, which are added to the mesh while updating the active front. This process continues until no further points can be reached, and can be repeated with multiple ball radii to handle uneven sampling densities. Efficient spatial queries are typically supported using octree-based acceleration structures.

Kazhdan et al. [33] presented a surface reconstruction pipeline based on the Poisson equation. The method interprets oriented points as samples of a 3D indicator function's gradient and solves for an implicit function whose gradient best matches the vector field. A watertight mesh is then extracted as an isosurface from this function using an adaptive octree. This global approach makes it highly resilient to noise.

Schnabel et al. [34] proposed Efficient RANSAC, a generic and scalable shape detection framework for unorganized point clouds. Candidate shapes (planes, spheres, cylinders, cones, tori) are generated from minimal point sets via hierarchical localized octree sampling. Promising candidates are scored using lazy evaluation on small random subsets and refined with full inlier evaluation. Accepted shapes are least-squares refitted for geometric accuracy, with inliers removed iteratively until no significant shapes remain. The approach is robust to high noise and outliers, detects

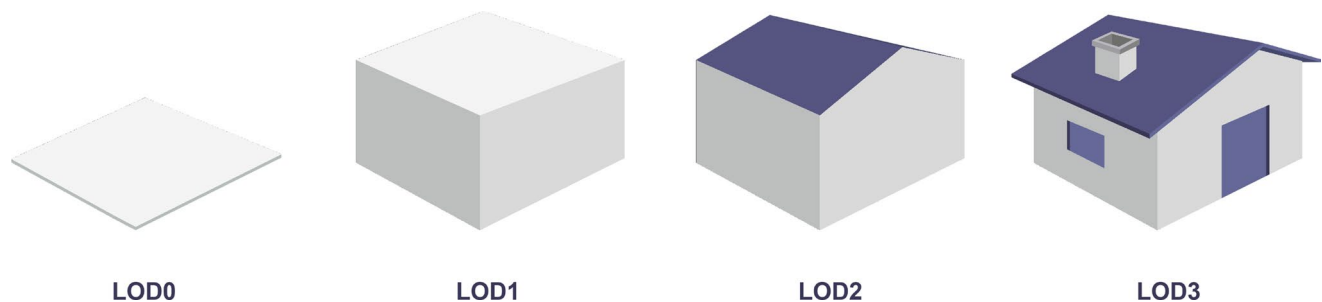


Fig. 5 Comparison of building levels of detail LOD-models in the standards, as adapted from Biljecki et al. [26] (author's own source)

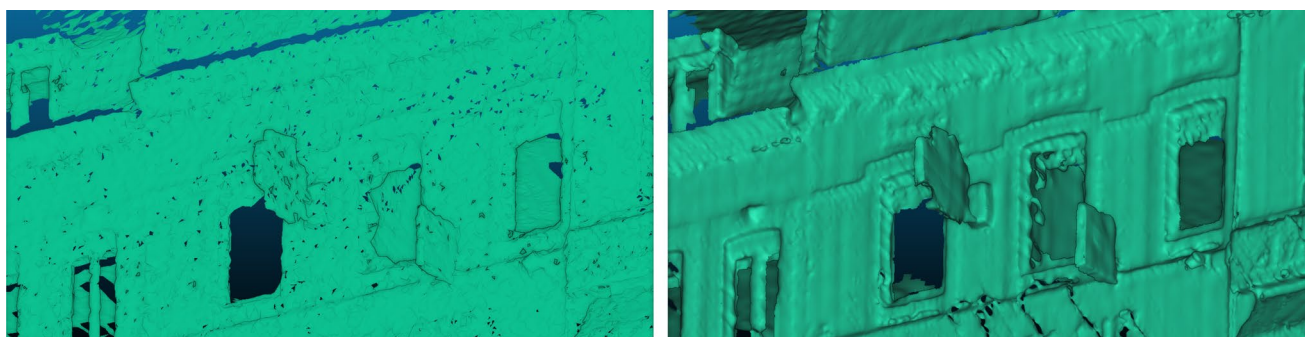


Fig. 6 Comparison of the ball Pivoting Algorithm [32] and Neural kernel surface Reconstruction [35] on the NPM3D/Paris-Lille dataset (Lille-2 tile) [12] (author's own source)

Table 3 Comparison of generic surface reconstruction methods

Method	Input	Output	Building Blocks
Marching Cubes [27]	3D volumetric grid data (voxels)	Triangulated isosurface mesh	1. Subdivide the grid into cubes. 2. Classify corners using iso-value. 3. Use a lookup table to generate a triangle mesh. 4. Compute normals for shading.
Ball Pivoting Method [32]	Oriented Point Clouds	Triangle mesh (orientable manifold)	1. Select the seed triangle. 2. Pivot ball to find neighboring points. 3. Add a triangle and update the active edge front. 4. Repeat with different ball radii to handle uneven sampling.
Poisson Surface Reconstruction [33]	Oriented Point Clouds	Watertight triangulated surface model	1. Compute the indicator function from point normals. 2. Solve the Poisson equation to get an implicit function. 3. Extract isosurface using Marching Cubes.
Efficient RANSAC [34]	Unorganized point clouds (optional normals)	Detected geometric primitives (planes, spheres, cylinders, cones, tori) with point sets	1. Preprocessing (optional filtering, normal estimation). 2. Minimal point set sampling using an octree. 3. Lazy candidate scoring on subsets. 4. Full inlier evaluation and selection. 5. Iterative inlier removal. 6. Least-squares refitting for accuracy.
Neural Kernel-Surface Reconstruction (NKSR) [35]	Point clouds with normals	3D implicit surface (mesh extracted via Dual Marching Cubes)	1. Learn sparse voxel hierarchy with features. 2. Represent surface as weighted sum of kernel basis functions. 3. Solve for kernel weights using gradient fitting and surface consistency.

multiple primitive types in a single run, and applies across domains from cultural heritage to mechanical parts.

More recently, learning-based methods like Neural Kernel Surface Reconstruction by Huang et al. [35] have emerged. They present an end-to-end pipeline that learns a 3D implicit surface as a weighted sum of learned, compactly supported kernel basis functions. These functions are centered on a sparse voxel hierarchy predicted by a sparse CNN.

While these methods are effective for generating meshes of arbitrary objects, they often fall short when applied to scenes of an urban environment composed of multiple objects. A key limitation is their inability to incorporate domain-specific priors such as planarity and orthogonality, which are essential characteristics of man-made structures. For instance, the global smoothing behavior of Poisson reconstruction tends to blur sharp architectural details like eaves and corners. Ball Pivoting, on the other hand, is highly sensitive to non-uniform point densities, often resulting in holes along sparsely scanned façades. In the case of Neural Kernel Surface Reconstruction (NKSR), although it

performs well on synthetic or object-centric datasets, its performance degrades with MLS data, likely due to a domain gap between training data and real-world urban scans. A visual comparison of some outputs is shown in Fig. 6, and a detailed comparison is given in Table 3.

3.3 Road Reconstruction Pipeline

While traditional surface reconstruction methods provide a generic mechanism for mesh generation, they often fail to capture the structured and topologically consistent nature of urban road networks. Roads play a foundational role in organizing city geometry and can act as geometric priors for downstream building reconstruction tasks. By explicitly modeling road geometry, we can improve global alignment, ensure consistent ground-level references, and reduce ambiguity in façade placement. This is especially useful in MLS scans where occlusions and clutter often degrade the quality of building outlines.

There are three critical challenges with using generic surface reconstruction techniques for road reconstruction.

Firstly, these methods struggle to handle large flat surfaces, discontinuities at intersections, and the varying point densities common in urban road scene point clouds. Secondly, they are often optimized for localized mesh fitting rather than reconstruction of large-scale structured surfaces in the form of networks. Thirdly, they lack semantic understanding and do not account for the unique surface characterization of road surfaces, curbs, sidewalks, and obstacles, which is needed for 3D road reconstruction. To overcome these limitations, several works have proposed dedicated road surface extraction and modeling pipelines, many of which target MLS data due to its high resolution and alignment with road geometry.

In practice, the road reconstruction pipelines use modified variants of the generic surface reconstruction methods. There are two types of pipelines depending on their inclusion of a *semantic segmentation* step:

- **Without Semantic Segmentation:** These pipelines directly use already-segmented road points or specific road scans for reconstruction.
- **Integrated with Semantic Segmentation:** These pipelines include a segmentation step to determine the road points before the reconstruction step.

A few examples of both types are listed below, and are summarized in Table 4.

3.3.1 Without Semantic Segmentation

Oude Elberink and Vosselman [36] proposed a reconstruction pipeline for complex, multi-level road interchanges using ALS LiDAR point clouds and 2D topographic maps. It involves point cloud segmentation into smooth surface patches via a surface-growing algorithm, merging of adjacent road polygons using Hough-transform-based local direction estimation, and correct road level assignment to points, through fitted planes. 3D boundaries are reconstructed by fitting planes to assigned points, and adjacency rules handle merging or step edges. This integration of LiDAR and map data addresses occlusions, achieves high vertical accuracy (10–15 cm), and scales to complex interchanges with up to four distinct height levels.

McElhinney et al. [37] presented initial results from the EURSI mobile mapping project using LiDAR and GNSS navigation data. Their pipeline extracts road cross-sections using vehicle trajectory data and fits 2D cubic splines to identify road boundaries. The spline geometry is filtered based on attributes such as elevation, pulse width, and intensity, providing lightweight polylines that represent road edges and centerlines.

Kumar et al. [39] designed an automated pipeline for extracting road edges from terrestrial mobile LiDAR data. The method constructs raster surfaces from attributes such as elevation, pulse width, and reflectance. Edge detection is performed using hierarchical thresholding and Canny filters. A combination of Gradient Vector Flow (GVF) and balloon energy snakes is then used to iteratively fit smooth

Table 4 Comparison of road surface reconstruction pipelines

Method	Input	Output	Building Blocks
Oude Elberink and Vosselman [36]	Point Cloud (ALS) and 2D Topographic Maps	3D road surface models, multi-level roads	1. Segment surfaces into smooth patches. 2. Merge road polygons via Hough-based map-growing. 3. Assign points to roads; fit planes to boundaries. 4. Merge or step edges using rules.
McElhinney et al. [37]	LiDAR and GNSS	Road cross-sections and splines	1. Cross-section extraction 2. Cubic spline fitting 3. Filtering via elevation and intensity
Hervieu and Soheilian [38]	Point Cloud (MLS)	Parametric road and pavement models	1. Compute θ feature to detect curbs. 2. Estimate curbs via predictive model. 3. Slice roads; fit polynomial surfaces via RANSAC.
Kumar et al. [39]	Point Cloud (MLS), GNSS/INS	3D polylines of road edges	1. Raster surfaces from LiDAR attributes 2. GVF and balloon snake models for contour fitting
Garach et al. [40]	LiDAR and Navigation data	Parametric road alignments (splines)	1. Pavement marking extraction 2. Spline segmentation 3. Curvature classification
Zhang et al. [41]	Point Cloud (MLS)	3D highway curve geometries	1. Split MLS data via trajectory. 2. Extract road surface; detect markings. 3. Fit clothoid curves to marking hulls.
Katkoria and Srevalsan-Nair [42, 43]	Automotive LiDAR and vehicle trajectory	Triangulated road surface mesh	1. Ground segmentation using RFC 2. RANSAC for edge fitting 3. Delaunay tetrahedralization
Wang et al. [44]	MLS, TLS, ALS	BIM-ready road models	1. SCF-Net segmentation 2. Voronoi, Alpha shape for layout 3. DEM and slope estimation
Davletshina et al. [45]	MLS point clouds (XYZ, RGB, normals, eigenvalues, optional GPS)	3D meshed, coloured, and semantically labelled geometric foundation models for road digital twins	1. Data partitioning (normals, eigenvalues, trajectory-based) 2. PointNet++ semantic segmentation (context-/location-aware) 3. DBSCAN clustering-based reclassification 4. Intensity-based line marking segmentation 5. Poisson surface reconstruction

road boundaries. The output is a pair of 3D polylines representing the left and right road edges.

3.3.2 Integrated with Semantic Segmentation

Hervieu and Soheilian [38] introduced a semi-automatic road and pavement modeling workflow using high-density MLS point clouds. An angular distance-to-ground-normal feature map is computed to highlight vertical structures such as curbs, which are then detected and tracked via a Kalman filter-inspired prediction/estimation model. Perpendicular street slices are generated, and RANSAC-based polynomial fitting is applied within each slice, using a quadratic polynomial for standard roads, and a cubic one for intersections, to model continuous road and pavement surfaces. The method achieves centimeter-level accuracy and adapts to varying road geometries without prior width assumptions.

Garach et al. [40] presented a spline approximation algorithm for extracting road alignments from LiDAR and navigation data. Pavement marking segmentation is first implemented using intensity, color, and scan angle thresholds, and used to determine the road axis. The road axis is further simplified using mean filtering, and curvature segments are extracted using variational spline fitting. Each segment is then classified into straight, clothoid, or circular curves, producing a parametric road alignment model.

Zhang et al. [41] reconstructed 3D parametric highway curves using vehicle trajectory data and additional scalar fields from point clouds. The trajectory is used to segment the MLS point cloud. Road surfaces are extracted via smoothness-based region growing, while road markings are detected using intensity variance thresholding and clustering. Hull points of marking clusters, that are extracted from alpha-shapes, are then fitted with robust parametric models (Mean Measure Estimator) for standard highway curve types, namely, horizontal curves for line, circle, and spiral, and vertical ones for line and parabola. The method achieves sub-percent relative error in curve parameter estimation.

Katkoria and Sreevalsan-Nair [42] introduced RoSELS, a triangulated surface extraction method for straight road segments from sequential automotive LiDAR scans. It uses handcrafted features and a Random Forest classifier for ground segmentation, followed by RANSAC-based edge detection and constrained Delaunay tetrahedralization for surface construction. An extension to RoSELS for curved road segment extraction [43] used ribbon extrusion from the vehicle trajectory information. This allows for better continuity across segmented frames and enables modeling of non-linear road structures.

Wang et al. [44] proposed a geometric information extraction framework that supports MLS, TLS, and ALS data. An improved SCF-Net with local and global

contextual encoders is first used for semantic segmentation. The extracted road points are processed using Voronoi diagrams and alpha-shape algorithms to compute boundaries and centerlines. Geometric attributes, such as slope and alignment, are derived and exported in a BIM-compatible format.

Davletshina et al. [45] presented an automated and generalized hybrid pipeline for automating the road geometry extraction for a digital twin using context- and location-aware segmentation. The datasets of interest are MLS point clouds, with RGB, normals, and eigenvalue features, with optional GPS metadata, e.g., KITTI-360 (urban and suburban roads in Germany) [46] and Digital Roads (UK highways, DR1, and DR2) [47]. Computed normals and eigenvalues are used for point cloud partitioning along the scanner trajectory. PointNet++ is used for multi-class semantic segmentation for primarily road assets, with an optional location-aware module using GPS priors and a binary PointNet++ classifier for accurate segmentation of rare or small assets. DBSCAN is used for clustering and reclassification for label consistency. Road surfaces and line markings are further extracted via an unsupervised module combining intensity filtering and clustering. Finally, watertight 3D road geometry is reconstructed using Poisson surface reconstruction. Context-aware segmentation improves accuracy and efficiency, especially in asset-specific detection.

3.4 Building Reconstruction Pipeline

Similar to road reconstruction pipelines needing semantic segmentation before surface reconstruction, building reconstruction pipelines require both semantic and *instance segmentations*. Section 3.1.1 gives an overview of different LOD models along with the visualizations, which are defined in detail by [26]. The LODs give an impression of the characterization of the reconstructed buildings. Most automated reconstruction methods target LOD1 or LOD2 due to the trade-off between complexity and data availability, especially from ALS or MLS. The two classes of building reconstruction methods, based on their modeling principles and data dependencies and as illustrated in Fig. 7, are:

- **Primitive- and Extrusion-based Modeling:** Primitive-based models extract geometric primitives (e.g., planes, cuboids, cylinders) in the point cloud and assemble them into building models, often leveraging roof segmentation. Some of the primitive and geometry-based models also cover extrusion-based techniques as an intermediate step. Extrusion-based approaches rely on 2D topographic maps or footprints and apply heuristic rules for height estimation and surface generation. Hence,

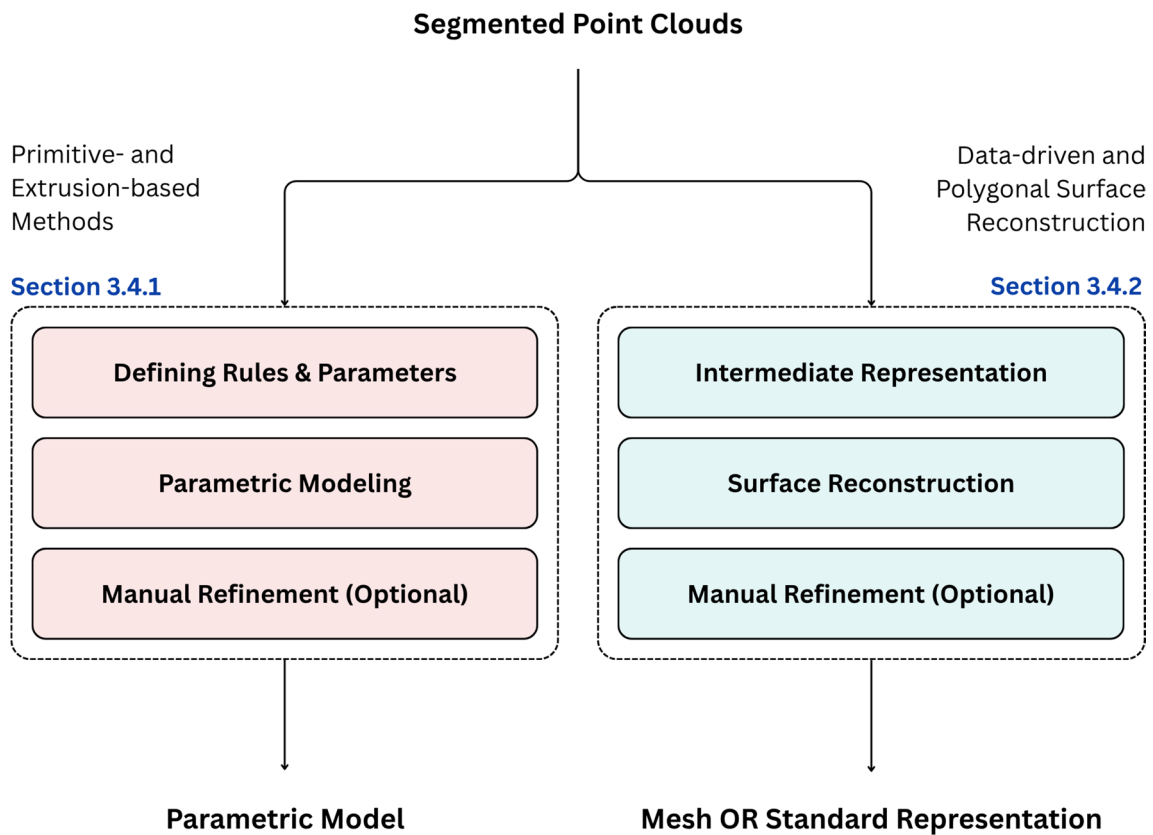


Fig. 7 Reconstruction pipeline applied to segmented point clouds, illustrating two reconstruction strategies such as primitive- and extrusion-based parametric modeling, and data-driven polygonal surface reconstruction. (author's own source)

extrusion-based methods come under the larger class of rule-based methods.

- Data-driven and Polygonal Surface Reconstruction:** In these methods, the building geometry is extracted directly from the point clouds, as exemplified in diverse point cloud applications, e.g., surface fitting, face generation, etc.

A few examples of both classes are listed below, and are summarized in Tables 5 and 6. A complementary classification based on modeling methodology has been proposed in [25] [1], where building reconstruction approaches are categorized as data-driven, model-driven, hybrid-driven, and semantic-driven. However, in this work, we adhere to a classification based on modeling principles, data dependencies, and reconstruction complexity.

One major point to note is that building pose estimation is inherently handled within most reconstruction pipelines, where the dominant orientation and spatial extent of each building instance are implicitly recovered as part of the processing. This provides the geometric basis for subsequent regularization. Dominant directions are typically obtained using classical geometric methods such as the Hough transform [48], RANSAC-based line fitting [58], or

PCA on segmented point clusters [60, 75]. These estimated orientations are then used to regularize footprint boundaries [49] and align primitive assemblies, ensuring consistent structural directions in the reconstructed geometry. In urban environments where orthogonal façade layouts are common, Manhattan-world assumptions [56] are often applied to further constrain orientation estimation, reducing ambiguity during both footprint extraction and roof plane fitting. When a road network is available as a structural prior, building orientations can additionally be aligned with road centerlines and boundary edges, which act as geometric axes of the street layout, thereby improving spatial consistency across the reconstructed scene [13].

3.4.1 Primitive- and Extrusion-Based Methods

Vosselman et al. [48] introduced one of the earliest primitive-based approaches for 3D building reconstruction from ALS data and footprints, where buildings are instantiated from footprints and roof planes are extracted using the Hough transform. Coarse geometry is obtained by fitting predefined roof primitives, with residual points refining finer structures such as dormers and chimneys. A closely related footprint-centric pipeline was later proposed by Zhang et al. [49],

Table 5 Comparison of primitive- and extrusion-based methods

Method	Input	Output	Building Blocks
Vosselman et al. [48]	Point clouds (ALS), 2D ground plans	LOD2 3D building models	1. Segment footprint points from cloud. 2. Extract planes via Hough transform. 3. Estimate height; fit roof primitives. 4. Refine with residual points for smaller structures.
Zhang et al. [49]	Point clouds (ALS)	Regularized 2D building footprints	1. Segmentation via progressive morphological filter. 2. Region-growing with plane fitting. 3. Boundary tracing and simplification. 4. Direction-based refinement.
Poullis et al. [50]	Point Clouds (ALS) and Aerial Images	Photorealistic textured 3D city models	1. Preprocess LiDAR via resampling, filtering, and ground/non-ground segmentation. 2. Detect buildings; extract/linearize roof boundaries; fit parameterized primitives. 3. Register imagery; generate seamless composite textures. 4. Integrate textured models into the final environment.
Dorningrand Pfeifer [51]	Point Clouds (ALS)	3D building models with regularized geometry	1. Extract buildings from ALS via segmentation/classification. 2. Derive alpha shape footprint from roof data; regularize via angular constraints. 3. Model roofs and vertical walls; regularize corners by merging close intersections.
Lafarge et al. [52]	Single DSM	3D building model	1. Extract 2D footprints as connected quadrilaterals (auto/semi-auto). 2. Fit parametric roof models from a predefined library via Gibbs model. 3. Optimize block configuration using RJ-MCMC with simulated annealing. 4. Generate final 3D model.
Zhou andNeumann [53]	Point clouds (ALS)	3D urban model	1. Stream LiDAR data for large-scale processing. 2. Classify ground, vegetation, and buildings via SVM. 3. Segment building patches using streaming clustering. 4. Generate terrain model; fit planes; snap boundaries to principal directions for watertight models.
Kada and-McKinley [54]	Point Clouds (ALS), Footprint	LOD2 3D building models	1. Decompose footprints into nonintersecting quadrangular/rhomboid cells. 2. Match cell normals to standard roof faces from a library. 3. Estimate eave/ridge heights; connect cells with junction primitives. 4. Manual refinement for complex cases.
Poullis andYou [55]	Point Clouds (ALS), Aerial imagery (Optional)	Lightweight, watertight polygonal 3D building models	1. Convert point cloud to XYZ map; resample and fill holes. 2. Segment via normal-based probabilistic clustering; remove non-building points. 3. Extract/simplify boundaries; fit roof plane via RANSAC. 4. Extrude footprint to the roof plane for a watertight model.
Vanegas et al. [56]	Segmented Point Clouds (TLS)	Manhattan-world volumetric models	1. Classify points into local shape types under the Manhattan-world assumption; refine classifications. 2. Cluster coplanar points; triangulate to form building faces. 3. Subdivide into AABBs; derive volumes via ray casting. 4. Smooth, simplify, and consolidate to the final polygonal model.
Zhou andNeumann [57]	Point clouds (ALS)	2.5D regularized building models	1. Extract planar roof patches via region growing; refine using global slope/position regularities. 2. Generate/refine boundary segments via height and position alignment. 3. Add façades, prune patches, and assemble into 2.5D model.
Chen et al. [58]	Point clouds (ALS)	Watertight 3D polyhedral building models	1. Separate ground/non-ground via progressive morphological filter; classify with SVM. 2. Segment rooftops using improved RANSAC with normal checks. 3. Extract/regularize boundaries via Voronoi and alpha shapes; align to dominant directions. 4. Simplify with Douglas–Peucker; generate a watertight model via constrained triangulation.
Awrangjeband Fraser [59]	Point clouds (ALS), DEM (Optional)	Extracted roof planes and boundaries	1. Ground removal and building mask generation. Followed by an eigenvalue-based coplanarity check. 2. Region growing for roof segmentation. 3. Merge/refine planes and polygonize outlines.
Awrangjeb [60]	Pre-segmented ALS building point clouds	Regularized 2D building footprints	1. Boundary from Delaunay triangulation. 2. Remove long edges; form closed loops. 3. Corner detection via curvature/voting. 4. Least-squares line fitting; enforce principal directions.
Chen et al. [61]	Point clouds (ALS)	Multi-LODs watertight polyhedral rooftop models	1. Enhanced PDC segmentation of planar facets. 2. Voronoi subgraph boundary tracing. 3. Boundary regularization; constrained triangulation.
Ledoux et al. [14]	2D polygons, ALS (optional)	LOD1 watertight buildings	1. Semantic lifting 2. Extrusion 3. Stitching
Zhang et al. [62]	Point clouds/photogrammetry, DEM	LOD2 polyhedral models	1. Primitive classification 2. Solid fitting with PMD 3. Superstructure clustering
Peters et al. [9]	2D footprints, ALS	LOD1.2 to LOD2.2 3D models	1. Roof segmentation 2. Partitioning and extrusion 3. Graph-cut assignment
Huang et al. [15]	Point clouds, optional footprints	Watertight lightweight polygonal model	1. Footprint vectorization 2. Polyline fitting 3. Hypothesis-selection and roof preference energy
Lewandowicz et al. [63]	ALS pre-segmented structures	LOD2/LOD3 building models (rotational surfaces)	1. Project the point cloud to the XY plane (2D footprint) and estimate the gravity center. 2. Extract vertical cross-section (density-based slice) and generate semi-cross-section. 3. Revolve around the vertical axis (cylinders and frustums) and generate surfaces using X, Y, Z matrices.

Table 6 Comparison of data-driven and polygonal surface reconstruction methods

Method	Input	Output	Building Blocks
Nan and-Wonka [64]	Point clouds with normals	Manifold polygonal model	1. Plane detection 2. Face generation via selection terms
Zhang andZhang [65]	ALS, TLS point clouds	Lightweight 3D building models with arbitrary roofs and vertical walls	1. Deep reinforcement learning for building classification 2. Edge-aware resampling for consolidation 3. 2.5D dual contouring for mesh reconstruction
Zhang et al. [66]	ALS, TLS point clouds (urban-scale)	Lightweight 3D building models with complex roofs	1. Handcrafted feature extraction (intensity, normals, spin images, etc.) 2. Sparse autoencoder for feature encoding 3. ReLu-NN for classification 4. Edge-aware repairing 5. 2.5D dual contouring for reconstruction
Chen et al. [67]	LiDAR, RGB point clouds	Semantic clusters and reconstructed 3D scenes with aligned CAD models	1. Graph-based edge segmentation with MLP 2. PointNet for object classification 3. Rule-based component merging 4. CAD model retrieval and alignment
Wysocki et al.[68]	Pre-existing 3D city models (CityGML LOD2/LOD3), MLS/ALS point clouds, HD Maps (OpenDRIVE)	Refined and semantically enriched 3D city models in CityGML LOD2/LOD3, Datasmith export	1. Clipping/filtering (lasground) 2. Coverage analysis, RANSAC segmentation 3. Screened Poisson reconstruction and mesh simplification 4. Semantic enrichment (object/attribute detection)
Pan et al. [69]	TLS point clouds, UAVDP	Semantic BIM models, integrated into Cesium-based Heritage Digital Twin	1. Data acquisition (UAVDP, TLS) 2. KP-SG segmentation (11 categories) 3. Scan-to-BIM in Revit/Civil 3D 4. Semantic enrichment (heritage attributes) 5. Cesium-based twin integration
Liu et al. [70]	Point clouds and semantic segments	Polygonal mesh (LOD2 like)	1. SparseCNN and Transformer mesh generator
Ding et al. [71]	Incomplete building point clouds (UAV, MLS)	Watertight lightweight LOD2 building models	1. GAN-based completion with curvature attention 2. MLS smoothing 3. RANSAC plane/polygon extraction 4. Energy-based polygon selection
Xiao et al. [72]	High-resolution stereo imagery (with infrared band and red band) and airborne LiDAR point clouds	LOD1–LOD2 building models	1. Image-based edge and line extraction using a Sobel-based operator and a constrained Hough transform. 2. Building point separation from LiDAR using nDSM height thresholds and NDVI-based vegetation removal. 3. Coarse building boundary generation using concave hull construction. 4. Roof contour extraction via buffer-based removal of non-building image line segments and equidistant mapping of LiDAR boundary points for edge validation and regularization. 5. Structured 3D reconstruction through roof elevation interpolation and local ground plane fitting.
Liu et al. [73]	Point clouds (XYZ, RGB, intensity)	3D wireframe models (parameterized edges)	1. Transformer point encoder 2. Edge attention generation 3. Noise-to-wireframe diffusion 4. Edge-aware denoising 5. DDIM inference 6. NMS, DBSCAN post-processing
Bui et al. [74, 74]	ALS/UAV LiDAR point clouds, orthophotos	3D building models (bounding-box-based)	1. UAV photogrammetry (orthophotos, point clouds) 2. Non-building filtering (CSF, U-Net, height rules) 3. DBSCAN clustering and convex hull 4. Mobile LSD roof edge detection and refinement 5. Bounding box reconstruction with height from point clouds
Li et al. [75]	Backpack mobile LiDAR (MLS) point clouds	Vectorized, topologically consistent 2D building footprints	1. Multi-layer accumulated projection (MLAP) to generate occupancy maps. 2. Wall point extraction via PCA-based normal estimation and verticality filtering. 3. Deep line-segment detection using distance and angle field prediction. 4. Pseudo-gradient construction and LSD-based line extraction. 5. Structural chain-based merging, completion, and geometric regularization.

who automated footprint extraction through rasterization, progressive morphological filtering, and plane-fitting-based region growing. Raw footprints are traced from boundary points and refined using Douglas–Peucker simplification and dominant direction enforcement, enabling accurate reconstruction even for complex and oblique footprints.

Poullis et al. [50] expanded their work to a photorealistic reconstruction framework [10] to support rapid, large-scale urban modeling by integrating ALS data with multi-source imagery. Grid-based resampling, normal-based graph-cut filtering, and statistical ground separation are followed by

simplified roof reconstruction using symmetry constraints and a streamlined parameterization. The pipeline further incorporates visibility-based subdivision and occlusion recovery for efficient photo-texturing, enabling scalable, geometry-aware city-scale reconstruction. In contrast to template-based methods, Dorninger and Pfeifer [51] proposed a fully automated pipeline that operates directly on raw ALS point clouds. Using mean-shift clustering, planar segmentation, and alpha-shape-based footprint refinement under angular parallelism and orthogonality constraints, complex roof structures are reconstructed through plane

intersections and corner regularization without predefined roof templates.

Lafarge et al. [52] presented a structural reconstruction approach that fits parametric roof models from a predefined library to building components extracted as connected quadrilaterals from DSMs. Using a Gibbs model optimized via reversible-jump Markov Chain Monte Carlo (RJ-MCMC), the method exploits architectural regularities to model complex buildings, although it requires an intermediate transformation from LiDAR point clouds to DSMs.

To address scalability, Zhou and Neumann [53] introduced a streaming-based ALS reconstruction pipeline capable of processing city-scale datasets with limited memory. Buildings are classified using SVMs, segmented via agglomerative clustering, and reconstructed through plane fitting and boundary regularization aligned with principal directions, while terrain and building modeling are handled in a unified process without tile-edge artifacts. Similarly, Kada and McKinley [54] proposed a footprint-driven LOD2 reconstruction approach based on cell decomposition, where building footprints are partitioned into quadrilateral or rhomboid cells and roof types are assigned using a predefined library, with geometric continuity enforced through connecting primitives.

Vanegas et al. [56] introduced a Manhattan-world-based method for generating watertight building mass models from TLS data by classifying local primitives and performing volumetric modeling via ray casting, assuming dominant orthogonal directions and flat roofs. The approach is robust and efficient, with 10–60 s runtime for 10 million points. Zhou and Neumann [57] enforced global geometric regularities in a 2.5D ALS-based framework by aligning roof patches, ridges, and boundaries before adding vertical façades, improving geometric consistency of different combinations of roof-boundary interaction constraints without relying on static roof-type libraries.

Several works focused on robust roof and footprint extraction as modular components. Awrangjeb and Fraser [59] proposed a fully automated roof plane extraction method using eigenvalue-based coplanarity and region growing, achieving high correctness and completeness. This was extended by Awrangjeb [60] with a footprint extraction and regularization module based on Delaunay triangulation, curvature-driven corner detection, and principal direction enforcement, preserving detail even in low-density and complex scenes. Complementarily, Chen et al. [58] introduced a watertight polyhedral reconstruction pipeline combining SVM-based classification, improved RANSAC roof segmentation, Voronoi-based boundary extraction, and topology-aware regularization, while Chen et al. [61] further emphasized topological consistency through Probability

Density Clustering (PDC) and constrained triangulation to produce multi-LOD polyhedral models.

Ledoux et al. [14] presented an automatic pipeline that reconstructs LOD1 building models using 2D topographic polygons (e.g., as given in OpenStreetMap (OSM) [76] and optional point cloud data at country-scale, e.g., the Dutch Kadaster. Each polygon is lifted into 3D space based on class-specific height rules and stitched into watertight meshes.

Zhang et al. [62] proposed a framework that fits holistic primitive-based solids to roof surfaces using PointNet++ for classification. Point-mesh distances are minimized via SLSQP optimization, and secondary structures are added using DBSCAN and refinement heuristics. The method outputs CityGML-compliant LOD2 models with detailed roof structures.

Peters et al. [9] extended the pipeline [62] to generate LOD1.2, LOD1.3, and LOD2.2 models using ALS data and 2D footprints. Roof planes are extracted via region growing, and segments are regularized using graph-cut optimization. This pipeline scales to millions of buildings and can produce 3D base maps of the Netherlands, i.e., at country-scale.

Huang et al. [15] presented a pipeline that segments buildings using vectorized footprints, fits polylines via optimal transport, and reconstructs lightweight polygonal models using a hypothesis-selection framework adapted from the PolyFit algorithm, which is for polynomial regression. A dedicated roof modeling module favors common roof types via an energy term, enabling scalable and watertight reconstructions [25].

As a special case addressing non-planar and curved architectural elements (ornamental towers, turrets, and domes), which are often omitted or oversimplified by planar methods. Lewandowicz et al. [63] proposed a strategy that, rather than fitting planar facets, extracts a vertical cross-section through the footprint's gravity center and reconstructs the geometry by revolving a semi-cross-section around a vertical axis. The resulting surface is approximated using cylinders and conical frustums encoded in matrix form. This rotational primitive strategy extends the scope of primitive-based reconstruction beyond polyhedral geometry, enabling LoD2 and LoD3 modeling of curved architectural features directly from airborne LiDAR point clouds.

3.4.2 Data-Driven and Polygonal Surface Reconstruction

Nan and Wonka [64] proposed a RANSAC-based polygonal surface reconstruction framework that performs plane detection, pairwise intersection, and objective-based face selection to generate manifold, watertight building models with controllable complexity. The approach is robust to noise and well-suited for isolated building reconstruction.

Along similar geometric lines, Sreevalsan-Nair et al. [77] introduced a semi-automated tensorline-based method for gabled roof extraction from segmented ALS point clouds. Tensorlines derived from eigenvalue-based local descriptors are mapped to roof topology graphs via least-squares fitting, enabling roof plane generation with improved coverage compared to feature-graph-based approaches and Marching Triangles.

Wysocki et al. [68] presented a geometry-driven pipeline for the refinement and semantic enrichment of existing 3D city models, referred to as “plastic surgery.” The method integrates CityGML LOD2/LOD3 models with MLS point clouds, ALS data, and OpenDRIVE [78] HD maps. The pipeline involves clipping and filtering MLS points using existing city geometries and ground filtering, using `las-ground` from `LASTools` [79]. Wall extraction is performed through coverage analysis and RANSAC-based planar segmentation, followed by screened Poisson surface reconstruction and mesh simplification via Quadric Edge Collapse Decimation (QECAD). In addition to geometric refinement, semantic enrichment identifies new urban objects such as manhole covers, achieving reduced geometric deviation, compared to existing LOD3 models, while avoiding computationally expensive DL models.

Xiao et al. [72] proposed a fully automatic, data-driven building reconstruction pipeline that integrates high-resolution stereo imagery with airborne LiDAR data to extract high-precision roof contours and perform structured 3D reconstruction. The approach utilizes explicit geometric feature extraction, combining image-derived edge and line detection using an improved Sobel [80] based operator and constrained Hough transform [81] with LiDAR based height analysis using normalized Digital Surface Model (nDSM) and a Normalized Difference Vegetation Index (NDVI) for vegetation removal. Coarse building boundaries obtained from LiDAR are refined through a multi-stage denoising and data fusion process that removes non-building linear features and validates true roof edges via equidistant mapping. The resulting roof contours are then used to reconstruct structured LOD1 and LOD2 building models by interpolating roof elevations from LiDAR points and estimating ground levels through local plane fitting. Experiments on the ISPRS Vaihingen benchmark dataset [66] demonstrate high boundary accuracy and computational efficiency, although the multi-stage nature of the pipeline introduces several tunable parameters that may require careful adjustment across different urban scenes.

3.4.2.1 Learning-Based Reconstruction Pipelines Early learning-based pipelines focused on semantic parsing followed by geometric reconstruction. Zhang and Zhang [65] presented a deep reinforcement learning framework com-

binning 3D-CNNs, Deep Q-Networks, and residual recurrent networks for semantic parsing of ALS and TLS point clouds. Edge-aware resampling preserves sharp features, and 2.5D dual contouring generates meshes with arbitrarily shaped roofs while maintaining wall verticality. Experiments on Tianjin, ISPRS Vaihingen [82], and ETH Zürich datasets [83] show scalability across diverse scenes, though performance decreases on very small point clouds. Extending this direction, Zhang et al. [66] proposed a data-driven pipeline using handcrafted geometric features encoded via sparse autoencoders and classified with shallow neural networks. After denoising, 2.5D dual contouring produces watertight and lightweight building models with complex roof structures, while requiring fewer training samples than deep CNNs. Results on Vaihingen [82], Oakland 3D [84], Toronto [85], and Semantic3D [83] datasets show scalability to large urban scenes.

Hybrid learning-geometry approaches further integrate deep models with structural reasoning. Chen et al. [67] introduced a scan-to-3D pipeline that combines graph-based instance segmentation with a modified PointNet architecture for object classification, followed by Computer-Aided Design (CAD) model retrieval and alignment. The method uses LiDAR and RGB data, trained on the S3DIS dataset [86] and tested on real-world construction site scans. While achieving high instance segmentation accuracy, the method incurs increased runtime due to edge-based processing. In a different application domain, Pan et al. [69] proposed a DL-driven pipeline for semantic BIM reconstruction from heritage point clouds acquired via UAV digital photogrammetry (UAVDP) and TLS. Point cloud segmentation using the KP-Siamese Graph (KP-SG) network enables accurate LOD2 BIM generation, semantic enrichment, and digital twin visualization on the Cesium platform [87], although BIM modeling remains semi-automatic and site-specific.

Regarding DL-derived polygonal representations, Liu et al. [70] proposed an autoregressive transformer-based approach that directly generates polygonal building meshes by predicting vertex and face sequences from classified point cloud segments. Using SparseCNN embeddings, the model learns mesh structure directly from geometry and generalizes well across cities, even without explicit roof modeling. Complementarily, Liu et al. [73] introduced EdgeDiff, a diffusion-based framework for reconstructing 3D building wireframes from point clouds. After feature (XYZ, RGB, intensity) encoding of raw point clouds with a transformer-based encoder, an edge attention module that highlights edge-relevant regions. RGB is the color value at each point, in the case of a *colored* LiDAR point cloud dataset. A noise-to-wireframe paradigm diffuses ground-truth edges into noise during training, with an edge-aware denoising

process recovering structural wireframes. Denoising Diffusion Implicit Model (DDIM), with flexible control of edge density, is used for inference. Topology-preserving post-processing involves non-maximum suppression (NMS) and DBSCAN. EdgeDiff, which was run on the Building3D dataset [88] produced topology-preserving wireframes with a lower poly count than mesh-based methods.

Recent work has increasingly focused on robustness to incompleteness and multimodality. Ding et al. [71] presented a hybrid DL-geometric pipeline that completes incomplete point clouds using a Generative Adversarial Network (GAN) with curvature-aware attention, followed by RANSAC-based plane extraction and energy-based polygon optimization to reconstruct watertight LOD2 building models. Bui et al. [74] combined UAV orthophotos with LiDAR point clouds to extract roof edges from 2D imagery using the Mobile Line-Segmentation Detection (LSD) method and infer 3D geometry via height estimation, enabling efficient reconstruction from noisy data. Finally, Li et al. [75] proposed a DL-based method for building footprint extraction from backpack MLS point clouds, where a lightweight UNet-like network predicts geometric feature fields for LSD, which is implemented using a fast LSD algorithm [89]. This is followed by structure-aware optimization to enforce geometric regularity and topological consistency.

3.5 Summary

Most road reconstruction pipelines are designed in isolation, with downstream applications of visualization or infrastructure inspection. Several of them, especially the ones without an explicit semantic segmentation step, use additional datasets, such as topographic maps and scanner trajectories, to enable the segmentation. Many rely on handcrafted features or assume specific road conditions, making it difficult to generalize them across cities.

Owing to more definitive results in countable object detection, there is more work in building extraction methods than in the road extraction pipelines. While primitive- and extrusion-based pipelines often rely on predefined footprints from cadastral data or OSM, several approaches focus on deriving these inputs directly from point cloud data and generating outputs with detailed roof modeling at a large scale. Footprint extraction methods produce clean and regularized 2D outlines that can be extruded or paired with primitive roof models, while roof modeling techniques reconstruct accurate roof geometries from segmented points, enabling higher-LOD building representations. Together, these pre-processing and modeling steps form critical components of scalable building reconstruction workflows. The data-driven and polygonal surface reconstruction relies on DL models, increasingly, which use the dense LiDAR point

cloud data. These models can reconstruct complex geometries in buildings.

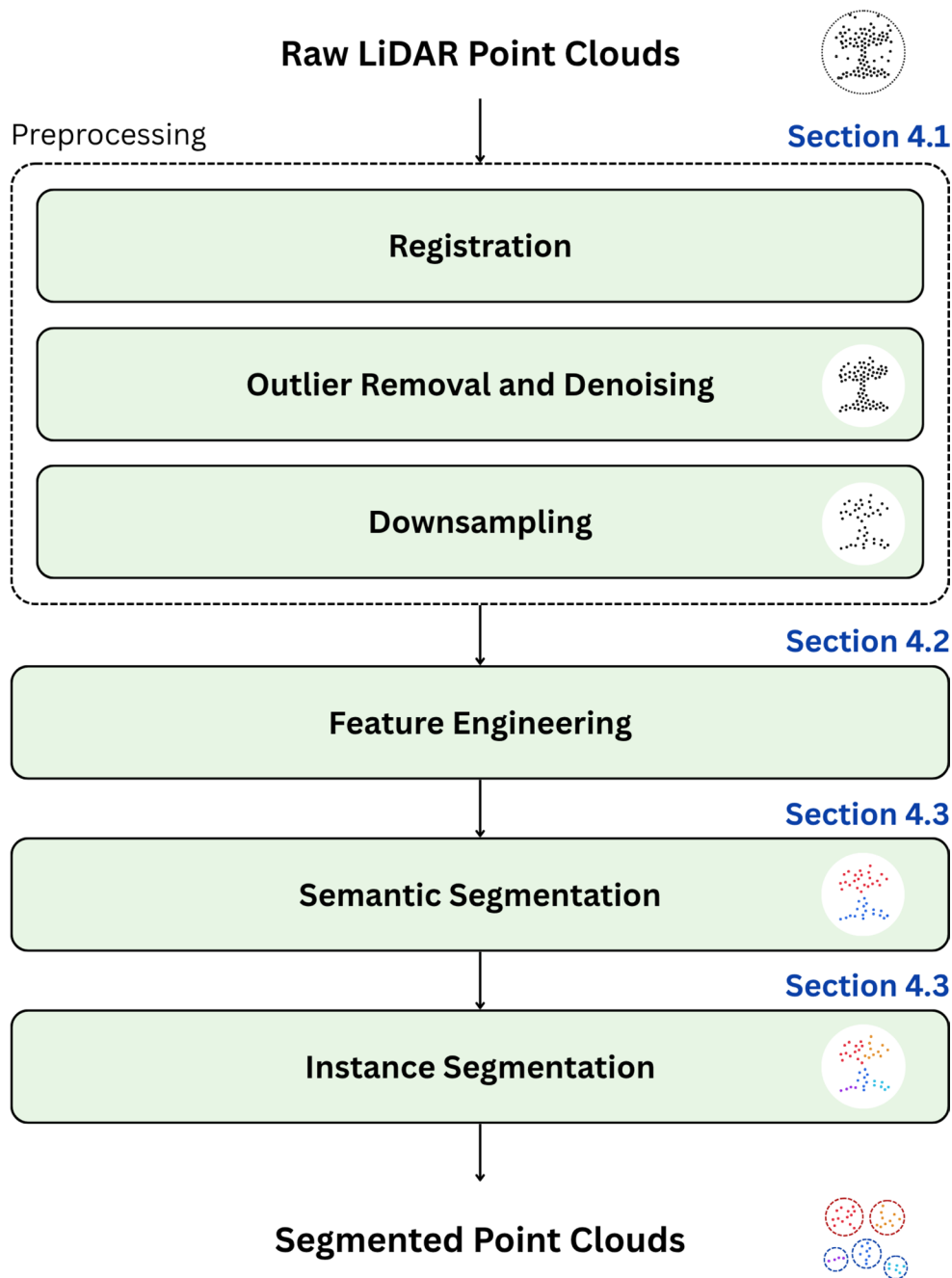
In most city-scale workflows, road reconstruction naturally precedes building modeling, since roads provide a stable spatial reference for the rest of the urban environment. Extracted centerlines, curb directions, and alignment geometry often act as reliable priors for downstream tasks such as pose estimation, footprint alignment, or bounding-box orientation. This becomes especially useful when building points are incomplete, rotated arbitrarily, or when no ground truth pose information is available. While not all buildings align perfectly with nearby roads, the dominant layout of planned urban areas often shows consistent orientation patterns, making road geometry a practical guide for large-scale reconstruction. Given limited work in integrated road-building reconstruction pipelines, there is scope for future work to develop unified city modeling pipelines that treat roads as both geometric and semantic priors, thereby enhancing overall reconstruction accuracy.

4 Background: 3D LiDAR Point Cloud Processing

To complete the discussion on the various reconstruction methodologies, it is important to outline the key LiDAR point cloud processing stages that lead to geometric reconstruction of semantic 3D city models. Point clouds provide positional information as they are unordered collections of points, typically represented by 3D coordinates (x, y, z) . They may also include additional metadata or scalar fields, either captured during acquisition or computed during pre-processing. Due to their large size, which ranges from millions to billions of points, efficient processing is critical for real-world applications and *downstream tasks*, namely, point cloud completion, object detection, and 3D reconstruction. This requires a careful selection of relevant data while discarding redundant or less useful attributes. The workflow for LiDAR point cloud processing needed for reconstruction is illustrated in Fig. 8.

Beyond the raw spatial coordinates, modern acquisition systems record a variety of point attributes such as intensity (laser return strength), scan angle (incidence angle of the laser beam), number of returns (multiple echoes from a single pulse), and pulse width (duration of the return signal). Not all of these attributes are directly useful for the downstream tasks, but they provide valuable contextual cues about surface properties and acquisition geometry. In many scenarios, point clouds may also be sparse or incomplete, motivating the interpolation of missing information or the derivation of additional features through feature engineering.

Fig. 8 Workflow illustrating the processing of LiDAR raw point clouds through outlier removal, downsampling, and geometric feature extraction, followed by semantic and instance segmentation to produce class wise structured point clouds.(author’s own source)



Apart from these attributes of the raw point clouds, higher-level geometric descriptors can be computed from local neighborhoods, such as eigenvalues and eigenvectors, planarity, verticality, and related measures [90, 91]. Such point-wise descriptors capture local geometric structure and are often required in downstream tasks. Hence, the feature extraction step substantially increases the dimensionality of each point, often tripling or quadrupling the number of attributes stored per point. This leads to an increase in space and time complexity for storage and computation, respectively. Consequently, efficient methods for managing and

processing point clouds remain a central challenge, given the scale and complexity of modern datasets, as listed in Tables 7 and 8.

Due to the complexity and scale of LiDAR data, several preprocessing operations are typically applied prior to any reconstruction or semantic modeling. The following subsections outline this standardized processing pipeline, covering preprocessing, feature extraction, semantic and instance segmentation, and an additional section covering data standard considerations such as CityGML, CityJSON, and BIM formats.

Table 7 A list of selected LiDAR benchmark datasets used for semantic segmentation

Dataset	Type	#Points [Point density]	#Classes	Classes
Vaihingen 3D [82]	ALS	1.2 M [4–8 ppm]	9	Low Vegetation, Impervious Surfaces, Tree, Roof, Shrub, Façade, Fence, Car, Powerline
TUM City Campus [92, 93]	MLS	41 M (v1), 1.7B (v2) [variable]	9	Unlabeled, Artificial Terrain, Natural Terrain, High Vegetation, Low Vegetation, Building, Hardscape, Artefact, Vehicle
Toronto-3D [85]	MLS	78.3 M [1K ppm]	8 (+1)	Road, Road Markings, Natural Objects (trees, shrubs), Buildings, Utility Lines, Poles, Cars, Fences, Unclassified
Paris-Lille-3D [12]	MLS	143 M [73.7K ppm]	50 / 10	Buildings, Roads, Sidewalks, Poles, Vehicles, Vegetation, Urban Furniture, and other urban classes
Semantic3D [83]	TLS	4B+ [variable]	8	Man-made Terrain, Natural Terrain, High Vegetation, Low Vegetation, Buildings, Hardscape, Scanning Artifacts, Vehicles
DALES [94]	ALS	505 M [50 ppm]	8	Ground, Vegetation, Buildings, Cars, Powerlines, and other urban and suburban features
H3D [95]	UAVLS	114 M [800 ppm]	11	Low Vegetation, Impervious Surface, Vehicle, Urban Furniture, Roof, Façade, Shrub, Tree, Soil/Gravel, Vertical Surface, Chimney

Table 8 A list of widely used LiDAR city-scale datasets and resources for semantic 3D city modeling

Dataset / Resource	Geographic Area	Specification	Miscellaneous Information
AHN (Actueel Hoogtebestand Nederland) [96]	Netherlands	Point Clouds (ALS)	High-resolution elevation dataset at country-scale; classes include unclassified, ground, building, water, and civil structures
3D BAG [9]	Netherlands	LOD1.2, LOD1.3, LOD2.2	Generated by combining BAG footprints with AHN elevation data; regularly updated; open access 3D building models
Project PLATEAU [97]	Japan	LOD1, LOD2	ALS-derived; 200+ cities, towns, wards; open digital commons supporting urban planning, digital twins, and large-scale semantic modeling
GlobalBuildingAtlas [98]	Global	LOD1, footprints and heights	2.68 billion LOD1 building instances; first open global dataset offering consistent 2D and 3D building data at individual building level
Random3DCity [99]	NA (Synthetic)	Multi-LOD	Procedurally generated dataset; buildings only; useful for controlled experiments and method validation

4.1 Preprocessing

The preprocessing steps include point cloud registration, outlier removal/denoising, and downsampling.

4.1.1 Registration

Usually, a point cloud is aggregated from multiple scans based on the snapshots taken by the LiDAR instrument. These input point clouds are subjected to *registration*, which is a process of aligning multiple scans to a common referential coordinate system using geometric or feature-based alignment techniques. This ideally involves an estimation of the transformation matrix between pairwise scans in a sequence. However, in practice, registration is performed by the acquisition system and its associated software, typically relying on metadata such as Global Positioning System (GPS), Inertial Navigation System (INS) measurements, scanner positions, and timestamp information.

Closer to this work, an earlier survey by Cheng et al. [100] focused on multi-platform, multi-temporal, and multi-angle LiDAR data in the photogrammetry and remote sensing domains. Same- and cross-platform registration involves different platforms for ALS, MLS, TLS, and SLS. *Coarse*

registration methods include point-, line-, and surface-features which are time-invariant. They are characterized by 3D keypoint operators (e.g., local surface patches, heat kernel signature, etc.), contours (e.g., building footprints and road networks), and surface-based methods (e.g., least squares surface and conjugate surface), respectively. *Fine registration* methods include iterative approximation, random sample consensus (RANSAC), normal distribution transform, or auxiliary data-based methods.

A recent comprehensive review done by Huang et al. [101] covers several techniques, including conventional ML and modern DL methods, for same- and cross-source registration. Optimization-based and feature-learning methods are used for both same- and cross-source registration, and end-to-end learning is used only for the same-source setting. Optimization-based models estimate the transformation matrix using Iterative Closest Point (ICP), graph-based, Gaussian Mixture Model (GMM)-based, and semi-definite methods. Feature-learning-based registration uses either volumetric data, point data, or both. End-to-end learning models use regression models or a combination of neural networks and optimization models. This survey predominantly focused on indoor and vehicle LiDAR point clouds, but is extendable to LiDAR scans of urban regions.

4.1.2 Outlier Removal and Denoising

The points from a registered LiDAR point cloud are samples of a set of surfaces where laser beams were incident upon and reflected from. While a majority of the points lie on the surface, there are a significant number of points that manifest from sensor noise, multipath returns, reflections from moving objects, or other acquisition artifacts. The additional points that are outside of the point distribution are referred to as *outliers*, which tend to be far from the objects in the scene. Points that lead to random errors near the *true surface*, fitted by the point cloud, are referred to as *noise*.

Outlier removal or filtering refers to the process of identifying and eliminating outlier points that do not correspond to the true underlying surface, based on an *approximate* statistical distribution of the surface points. Especially in this context, the outlier points provide no benefit to the reconstruction pipeline and instead degrade its performance.

There are geometric and statistical outlier removal methods routinely used for LiDAR point clouds. Geometric methods use point density, plane fitting, and local shape information to find outliers [102]. Statistical outlier removal (SOR) methods include outlier identification using spherical- or k -nearest local neighborhood of a given point [103, 104]. Carrilho et al. [105] proposed an adaptive statistical outlier filtering method that operates on local neighborhoods or smaller patches, rather than relying on a global histogram, to more effectively detect and remove outliers based on local point distributions. In addition, their work provides an extensive discussion of existing and widely used outlier removal techniques, including those implemented in *CloudCompare*, which are based on the widely used SOR approaches [106, 107].

The fast cluster statistical outlier removal (FCSOR) method [108] extends the classical SOR to address its high computational cost. The approach combines downsampling with outlier removal by dividing the point cloud into clusters and analyzing local point neighborhoods. Clusters with point counts exceeding the global average are discarded, reducing redundancy and significantly improving computational efficiency. This method was primarily developed for applications in unmanned ground vehicles (UGVs).

Denoising or noise removal refers to eliminating the points identified as noise from the point cloud. While outlier removal and denoising are interchangeably used for LiDAR point clouds, there are several exclusive denoising methods. For instance, principal component analysis (PCA) [109, 110] is extensively used for denoising LiDAR point clouds. Bilateral filters [111, 112] and learning models, e.g., unsupervised learning [113], have also been used for point cloud denoising.

4.1.3 Downsampling

Given the high density of modern point clouds, direct processing at full resolution is both computationally expensive and often unnecessary. Downsampling addresses this by reducing the number of points while retaining the essential geometric fidelity. A comprehensive study by Tzermia and Malamos [114] provides a detailed comparison of various downsampling techniques and serves as a useful reference for further discussion.

One commonly used technique is farthest point sampling (FPS), valued for its ease of use and reliability. A widely adopted variation is *FastFPS* [115], which leverages fast marching for incremental computation. The core idea is to begin with a randomly selected point and then iteratively add the point that lies farthest from all previously chosen ones. This process continues until the desired number of points is reached, ensuring that the sampled points are well distributed across the entire cloud.

Voxel grid sampling [116] is another simple and effective method for this purpose. It begins by voxelizing the point cloud and then selecting a representative point (typically the centroid) for each occupied grid cell. Associated scalar fields can be averaged and assigned to this representative point as the grid cell features. In this way, the number of points is reduced while the geometric structure of the object is preserved. A dynamic downsampling algorithm [117] builds on the idea of voxel-based filtering and addresses the issues of detail loss and the need for parameter tuning.

Poisson disk sampling (PDS) [118] is a progressive sampling process that ensures points are well spaced within the cloud. It enforces a minimum distance constraint so that no two points are closer than a specified threshold. This prevents clustering and provides a more even distribution of points, making it well-suited for preserving geometric details while reducing the number of points. Random sampling is also commonly used as a straightforward baseline, where points are selected randomly from the cloud until the desired number is reached.

Different from conventional subsampling methods, a multi-resolution point selection strategy is an alternative for massive point clouds [119]. This involves high-performance GPU rendering of large-scale point cloud data using a multi-way kd-tree hierarchy [120]. Here, each internal node in the hierarchy stores a fixed number, k , of representative points, which are determined using three simplification strategies—a feature-preserving normal deviation clustering, an entropy-based reduction that greedily minimizes angular and spatial information loss, and a lightweight k -clustering approach that refines an initial hash-based grouping. These methods operate directly on local point patches, producing compact yet feature-aware subsets.

4.2 Feature Engineering

The preprocessed point cloud is ready for the downstream tasks using computational methods and machine learning (ML). For using traditional ML models, handcrafted features are required to build a *feature vector* at each point. In the context of reconstruction, such features may be optionally incorporated, particularly when using data-driven or ML-based segmentation, and surface reconstruction methods.

Feature engineering refers to the process of deriving handcrafted features, i.e., descriptive attributes from raw point cloud, \mathcal{P} , to better capture local geometry, structure, and contextual information. While raw position coordinates provide the basic spatial structure, *derived features* are required to capture the *spatial locality* information. Thus, a few of these derived features from the position coordinates *describe* the geometry of the *local neighborhood* of each point. Hence, these features are referred to as the *local geometric descriptors* of the point. Local neighborhood of a point, $\mathbf{p} \in \mathcal{P}$, is defined as the set of points in the point cloud \mathcal{P} that satisfy a chosen *local neighborhood criterion*. This criterion describes the distance based on the *shape* of the neighborhood, or the rank of closeness/proximity. The latter is used in the definition of *k*-nearest neighborhood. The neighborhood in point cloud analysis can be spherical, cubical, or cylindrical, which are achieved using radial/Euclidean distance (L_2 norm) in 3D, Chebyshev distance (L_∞ norm), or radial/Euclidean distance in 2D projection to the x-y plane.

A widely used method for computing these point-wise descriptors involves the *covariance matrix* C , which captures the spread and orientation of points in a local neighborhood. C is also referred to as *structure matrix* in photogrammetry and remote sensing literature. Covariance matrix of point \mathbf{p} , $C(\mathbf{p})$, is computed with respect to its local neighborhood $\mathcal{N}(\mathbf{p})$ and its centroid $\bar{\mathbf{p}}$, as:

$$C(\mathbf{p}) = \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} t(\mathbf{q}) \cdot t(\mathbf{q})^T, \text{ where } t(\mathbf{q}) = (\mathbf{q} - \bar{\mathbf{p}}) \text{ and } \bar{\mathbf{p}} = \frac{1}{|\mathcal{N}(\mathbf{p})|} \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} \mathbf{q}.$$

The commonly derived geometric features at a given point, \mathbf{p} , are computed from the eigenvalues of its covariance matrix, $C(\mathbf{p})$, i.e., λ_i for $i = 1, 2, 3$, where λ_1 , λ_2 , and λ_3 are the major, middle, and minor eigenvectors, respectively. These features include linearity, planarity, scattering, eigenentropy, change of curvature, omnivariance, etc. [90, 91]. For point \mathbf{p} :

$$\text{Linearity}(\mathbf{p}) = \frac{\lambda_1 - \lambda_2}{\lambda_1}$$

$$\text{Planarity}(\mathbf{p}) = \frac{\lambda_2 - \lambda_3}{\lambda_1}$$

$$\text{Scattering}(\mathbf{p}) = \frac{\lambda_3}{\lambda_1}$$

$$\text{Omnivariance}(\mathbf{p}) = \sqrt[3]{\lambda_1 \lambda_2 \lambda_3}$$

$$\text{Anisotropy}(\mathbf{p}) = \frac{\lambda_1 - \lambda_3}{\lambda_1}$$

$$\text{Eigenentropy}(\mathbf{p}) = - \sum_{i=1}^3 \lambda_i \ln(\lambda_i)$$

$$\text{Sum-of-Eigenvalues}(\mathbf{p}) = \lambda_1 + \lambda_2 + \lambda_3$$

$$\text{Change-of-Curvature}(\mathbf{p}) = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$

These features are widely used for geometric classification of points, which further leads to the downstream tasks. By combining raw coordinates, scalar fields, and computed geometric features, point clouds can be represented using both local and global features. Also, each unique downstream task requires an optimal feature set.

As an example of optimal feature selection for a downstream task, a simple binary segmentation can be considered for road reconstruction. The segmentation task, here, classifies points into the road class and the non-road class, which is an aggregate of all other classes. At first glance, one might assume that the raw z -coordinate (height) is sufficient as a strong discriminative feature for the classification. However, in practice, this assumption does not hold true, as the actual data often includes height variations of structures in roads, e.g., ramps or slopes.

Example 2 A comparison of the rendering of a point cloud based on its height map, before and after filtering using a z -threshold, is shown in Fig. 9. Here, a simple z -value-based filtering proves ineffective because the feature is global rather than local.

Since the filtered point cloud still includes parts of other classes such as vegetation and buildings, and much of the ground data is missing, a more appropriate feature is a geometric one, namely, *Verticality* [121, 122], given by $V(\mathbf{p})$ at point \mathbf{p} with $\hat{\mathbf{z}}$ as the unit vector of the z -axis, and $\hat{\mathbf{e}}_3$ as the unit vector of the minor eigenvector of its local covariance matrix, $C(\mathbf{p})$:

$$\text{Verticality}(\mathbf{p}) = 1 - |\hat{\mathbf{z}} \cdot \hat{\mathbf{e}}_3|$$

Here, $\hat{\mathbf{e}}_3$ is the local surface normal, which is normal to the tangent plane fitted by the major and the middle eigenvectors, namely, $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$, respectively [28]. Verticality

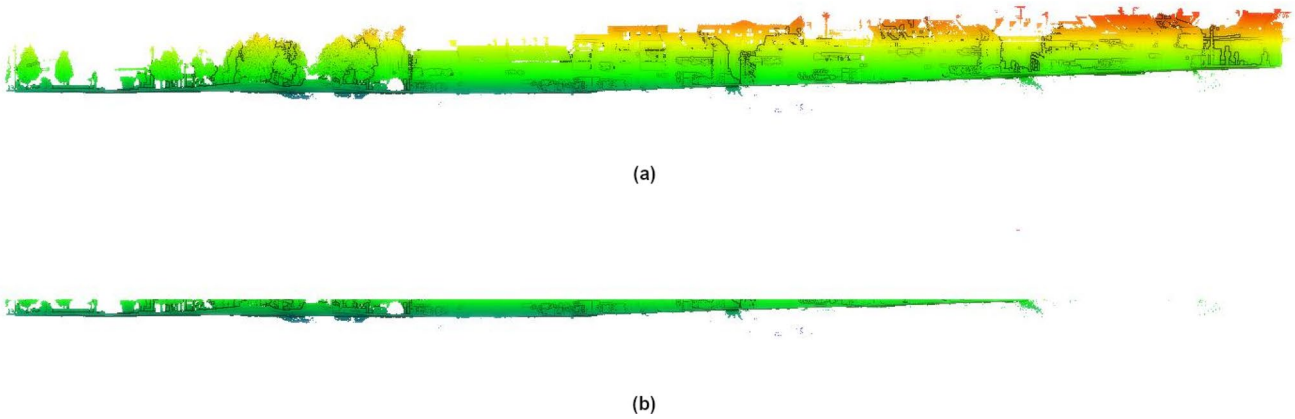


Fig. 9 Rendering of a tile of the NPM3D/Paris-Lille dataset [12] implemented as (a) the entire point cloud using the height (z) map, and (b) a filtered point cloud using a z -threshold, i.e., *ground filtering* (author's own source)

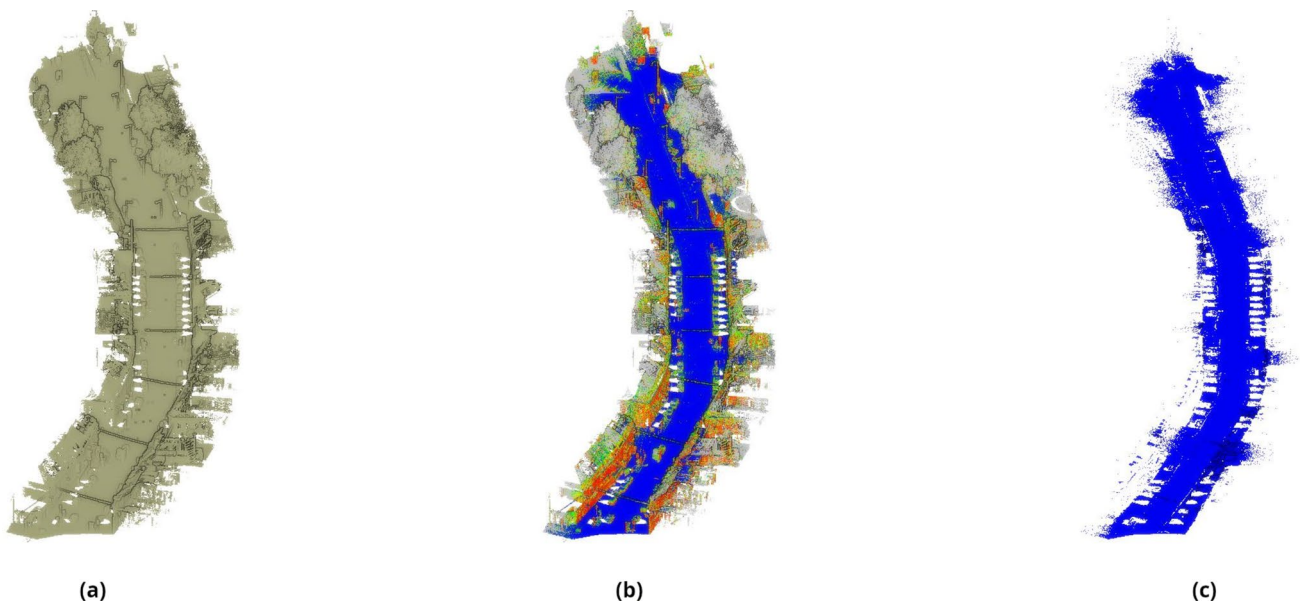


Fig. 10 The same tile in Figure 9 visualized as (a) the entire raw point cloud, along with (b) its computed verticality map, and (c) a filtered point cloud using a verticality-threshold (author's own source)

describes the degree to which the locally fitted plane of the local neighborhood of a point deviates from the horizontal plane at the point.

Example 3 The road and surrounding ground regions can be effectively separated using a simple threshold on the verticality values, as shown in Fig. 10.

The feature computation and visualizations for Figs. 9–10 were carried out using the *CloudCompare* tool [123].

Optimal feature set indirectly implies the selection of the *optimal* local neighborhood of a point to ensure that the computed covariance matrix accurately captures local geometric patterns. The suboptimal neighborhood sizes and shapes can either smooth out important details or amplify

noise. Both local and global geometric contexts are important: local features capture fine details essential for identifying small objects or structural variations, while global features encode broader spatial arrangements useful for understanding the scene layout. Optimal size of the local neighborhood has been widely determined using Shannon entropy function [124], where minimum entropy occurs at the *optimal* neighborhood size. This minimum entropy can be a global or local extremum value, and it implies a *stable* neighborhood at the optimal size.

4.3 Segmentation

By virtue of *spatial locality*, points in the local neighborhood tend to have the same class label, thus dividing the

point cloud into several *logical segments*. It is a foundational step in point cloud processing and plays a crucial role in further downstream reconstruction tasks [125].

There are different types of segmentations, based on the type of pointwise class labels, namely:

- **Semantic Segmentation** – The class labels are the names of the objects that the points are samples of, e.g., *Building, Road, Pedestrian, Car, Vegetation*, etc. This is the basic segmentation step to be implemented in a point cloud analysis pipeline, which provides the semantic context of the points.
- **Instance Segmentation** – The class labels are of the instances of the *countable objects*, e.g., *building, car, pedestrian*, etc. Here, the class label includes a unique index for the instance of the object that the point is a sample of. Hence, the outcomes are *Building-1, Building-2, Car-1*, etc. Instance segmentation does not include classes like *ground, vegetation, road*, etc., which are not considered countable. Since the object class has to be identified with the point samples of the concerned instance, instance segmentation is implemented after semantic segmentation [126].
- **Panoptic Segmentation** – This is a combination of semantic and instance segmentations. Thus, the labels are *Building-1, Building-2, Road, Vegetation, Car-1, Pedestrian-1*, etc.
- **Geometric Segmentation** – This is a class label of the geometric feature that the point is a sample of. Hence, the labels are *Line-, Point-, and Surface-type* features. Usually, the geometric segmentation, or geometric classification, is handled in the feature engineering step. The geometric classification can be either deterministic or probabilistic, depending on the binary or ratio value used for class membership [127]. Geometric segmentation is usually implemented before semantic segmentation, as the geometric features are used as handcrafted features, as explained in Sect. 4.2.

Semantic, instance, and panoptic segmentations can be considered to be different aspects of *3D object segmentation* [128]. 3D object segmentation provides appropriate subsets of point clouds that can be further used for obtaining fitting edges and planes for 3D reconstruction of object instances. It must be noted that instance segmentation is different from *3D object detection*, as the latter gives appropriate bounding boxes containing the subset of the point cloud, while the former gives a more compact shape outlining the silhouette of the object. Hence, instance segmentation [126] makes 3D reconstruction more efficient than object detection, especially since advanced surface reconstruction methods [35]

struggle to generalize across diverse outdoor scenes with varying densities and noise [25].

Ideally, panoptic segmentation provides the optimal subsets for 3D reconstruction, especially for buildings and roads. A recent survey on generic DL-based panoptic segmentation by [129] provides a few major techniques on point clouds as well. These techniques heavily rely on the preceding semantic segmentation.

In addition to these categories, recent work [130] explores fine-grained structural feature segmentation for reconstruction tasks. Unlike semantic or instance segmentation, such approaches identify geometrically meaningful features such as roof boundaries, fold edges, and planar regions. The authors propose a machine learning-based method that classifies roof points into these categories using handcrafted geometric features, providing an intermediate representation for downstream building reconstruction.

4.3.1 Semantic Segmentation in Practice

A simple and widely adopted approach is to perform semantic segmentation first, followed by instance segmentation via clustering methods (e.g., Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [131], region growing [132, 133], or connected component analysis [134, 135]) to distinguish between different object instances. In the case of building instance segmentation, this remains the dominant method in literature and practice. Therefore, this section focuses on semantic segmentation algorithms, while briefly touching upon their extensions toward panoptic tasks.

4.3.1.1 Representative DL Models for Point Cloud Semantic Segmentation Existing DL segmentation models developed for other domains do not directly support the unordered nature of point clouds, and their direct application is often inefficient. This challenge is commonly addressed through intermediate data representations. Comprehensive reviews by Liu et al. [136], Bello et al. [137], and Guo et al. [138] broadly classify existing data representation-based methods into three categories: *point-based* (direct processing of raw point sets), *grid-based* (voxel-based representations followed by convolutions), and *projection-based* (rasterization into one or more views followed by convolutional processing). Each category presents inherent trade-offs: projection-based methods may lose spatial information, voxel-based approaches are computationally expensive, and point-based models, while lightweight, often sacrifice accuracy.

A set of representative DL models for point cloud semantic segmentation is discussed here. Point-based segmentation using *PointNet* [139] and *PointNet++* [140] remains

widely used. PointNet performs per-point feature extraction combined with global feature aggregation, while PointNet++ additionally incorporates local neighborhood information. Thus, PointNet++ captures both local and global context.

Extending these ideas, Thomas et al. [141] introduced *Kernel Point Convolution (KPConv)*, a convolution operator designed specifically for point clouds. KPConv follows an encoder–decoder architecture with progressive downsampling and upsampling, replacing pooling operations with grid-based subsampling. Convolution kernels are defined in Euclidean space, with rigid and deformable variants used for small and complex point clouds, respectively. While rigid kernels maintain fixed kernel geometry, deformable kernels adapt their kernel point positions during training to better fit local geometric structure. The authors proposed two architectures based on these blocks, which are a *Convolutional Neural Network (CNN)* and a *Fully Connected Neural Network (FCNN)*, namely, KP-CNN and KP-FCNN architectures, respectively. They are used for classification and segmentation, respectively, achieving state-of-the-art performance on several benchmark datasets.

For large-scale point cloud segmentation, Hu et al. [142] proposed *RandLA-Net*, an efficient encoder–decoder architecture using random sampling as a constant-time density reduction strategy. To alleviate information loss, a Local Feature Aggregation (LFA) module combines Local Spatial Encoding, Attentive Pooling, and Dilated Residual Blocks. Shared *Multilayer Perceptrons (MLPs)* are used for initial feature extraction, while nearest-neighbor interpolation supports upsampling in the decoding phase, followed by fully connected layers with dropout for semantic prediction. RandLA-Net demonstrated strong performance on challenging LiDAR datasets such as Semantic3D [83], SemanticKITTI [143], and S3DIS [86], while maintaining high computational efficiency. However, its reliance on random sampling can reduce performance for minority classes and small objects.

Feature-Kernel Alignment Convolution (FKACov) by Boulch et al. [144] further improves point convolution by decoupling kernel weight estimation from spatial alignment. An alignment matrix is dynamically learned from local neighborhoods using permutation-invariant MLPs and max-pooling, while scale normalization and spatial gating help maintain robustness. The resulting residual encoder–decoder architecture achieves strong performance for both classification and segmentation tasks on benchmark datasets, e.g., S3DIS [86] and NPM3D [12].

Finally, for ALS-specific scenarios, Xiu et al. [145] introduced a self-supervised learning framework based on *Masked Auto-Encoders (MAE)*. MAE techniques, originally popularized in language modeling [146] and vision [147],

have recently been adapted to point cloud analysis [148]. As an example of using intermediate representations, BEV-MAE projects point clouds into a bird’s-eye-view (BEV) representation for segmentation [149].

5 LiDAR Datasets for 3D City Models

The availability of high-quality and large-scale datasets has played an important role in advancing research on 3D urban reconstruction. Public benchmark datasets not only enable reproducible experiments but also facilitate comparative evaluation across diverse reconstruction pipelines. However, datasets containing meshes or LOD-city models remain relatively scarce compared to raw point cloud datasets, which are primarily for semantic segmentation benchmarking. This section provides an overview of the key datasets that can be used for the development and validation of building and road reconstruction methods.

5.1 Semantic Segmentation Benchmark Datasets

The LiDAR point cloud datasets, benchmarked for semantic segmentation exclusively, tend to focus on well-defined areas such as street blocks, intersections, individual buildings, and, at times, city-scale. These datasets typically provide high-resolution point clouds with detailed semantic labels, making them valuable for developing and evaluating specific reconstruction modules in addition to semantic segmentation, such as footprint extraction or roof modeling. A selection of representative and widely used semantic segmentation benchmark datasets relevant to urban reconstruction is summarized in Table 7, which also provides the point density in points/m² (ppm).

This list does not include vehicle LiDAR datasets, also considered as MLS, such as KITTI [150], SemanticKITTI [143], KITTI-360 [46], etc., which are primarily benchmark datasets for semantic scene understanding.

5.1.1 Vaihingen 3D (V3D) Dataset

The V3D dataset [82], also referred to as the ISPRS 3D Semantic Labeling benchmark dataset, is one of the first high-quality ALS datasets made publicly available for a labeling contest. It has data collected from three different sites with characteristic differences, namely “Inner City,” “High Riser,” and “Residential Area.” These sites gave a variety of roof shapes, street width, vegetation density, etc.

5.1.2 TUM City Campus MLS Dataset

The TUM City Campus MLS dataset [92, 151] is a large-scale mobile LiDAR dataset acquired at the campus of the Technical University of Munich (TUM) in Germany. Version 1 consists of approximately 1.7 billion 3D points, while Version 2 includes over 2 billion 3D points, both with basic fine registration applied. The point clouds are provided in PCD format and include raw coordinates, echo amplitude, and scanner trajectory metadata, making the dataset highly valuable for research in semantic segmentation, change detection, large-scale urban reconstruction, and LiDAR processing.

5.1.3 Toronto-3D Dataset

The Toronto-3D dataset [85] is a mobile LiDAR Scanning (MLS) dataset acquired in Toronto, Canada, containing approximately 78.3 million points that cover four urban road sections of about 250 meters each. Provided in .ply format, each point has attributes such as position, RGB color, intensity, GPS time, scan angle rank, and semantic labels. It includes eight annotated classes, such as road, road markings, natural objects (trees and shrubs), buildings, utility lines, poles, cars, and fences, along with one unclassified category. All labels were manually annotated using CloudCompare [123], making Toronto-3D a high-quality and challenging benchmark for evaluating segmentation and reconstruction algorithms in urban environments.

5.1.4 Paris-Lille-3D Dataset

The Paris-Lille-3D dataset [12] is an MLS dataset acquired across the cities of Paris and Lille, France, designed to support research in large-scale urban scene understanding and reconstruction. It contains approximately 143 million points provided in .ply format and includes two annotation versions: a 50-class fine-grained semantic labeling and a 10-class simplified version for broader tasks. The dataset offers rich geometric and semantic diversity, encompassing complex urban structures such as buildings, roads, sidewalks, poles, vehicles, and vegetation, making it a widely used benchmark for evaluating segmentation, classification, and reconstruction algorithms.

5.1.5 Semantic3D Dataset

The Semantic3D dataset [83] is one of the largest publicly available Terrestrial LiDAR Scanning (TLS) benchmarks for point cloud classification and semantic segmentation. Acquired across various locations in Central Europe, the dataset contains over 4 billion 3D points stored as zipped

ASCII text files, representing both urban and rural environments such as farms, town halls, sports fields, castles, and market squares. The dataset defines eight semantic classes, such as man-made terrain, natural terrain, high vegetation, low vegetation, buildings, hardscape, scanning artifacts, and vehicles. Owing to its large scale, fine-grained labeling, and variety of scenes, Semantic3D remains a key benchmark for evaluating generalization and robustness in 3D semantic segmentation and reconstruction research.

5.1.6 Dayton Annotated LiDAR Earth Scan (DALES)

The DALES dataset [94] is a large-scale airborne laser scanning (ALS) dataset captured over the City of Surrey, British Columbia, Canada, across two days of acquisition. It comprises approximately 505 million 3D points, provided in .las, .txt, and .ply formats. The dataset includes eight semantic classes representing diverse urban and suburban features such as ground, vegetation, buildings, cars, and powerlines. DALES covers a wide range of land-use categories, including commercial areas (warehouses and office parks), high-rise urban zones (buildings over four stories), suburban neighborhoods (single-family homes), and rural regions (natural landscapes with sparse structures). Its extensive spatial coverage, balanced class distribution, and fine-grained manual annotations make DALES a valuable benchmark for evaluating large-scale aerial scene understanding and reconstruction workflows.

5.1.7 Hessianheim 3D (H3D)

The H3D dataset is an UAV laser scanning (UAVLS) dataset [95] captured over the village of Hessianheim, Germany, which was surveyed in three epochs during 2018–2019. The original paper covered only the first epoch, which was fully labelled. The full dataset with 114M points has a 50:10:40 split for training, testing, and validation, respectively [152]. The dataset also includes a textured mesh generated from oblique imagery.

5.2 Semantic 3D City Model Datasets Using LiDAR

Large-scale urban datasets provide extensive point cloud coverage of a city center and its suburbs. In some cases, these datasets also include mesh or LOD ground truths for evaluation. They enable testing and benchmarking of reconstruction pipelines for buildings and roads, supporting both geometric and semantic analysis at the city scale. Some key datasets relevant to semantic 3D city modeling are summarized in Table 8. The point cloud is available as .las and ASCII files, and the mesh is available as .obj, .mtl, and .jpg, to include the texture atlas.

5.2.1 Actueel Hoogtebestand Nederland (AHN) Dataset

The Actueel Hoogtebestand Nederland (AHN) dataset is a large-scale ALS survey covering the entirety of the Netherlands, providing high-resolution elevation information with five semantic classes: unclassified, ground, building, water, and civil structures. Building upon AHN, the **3D BAG dataset** by Peters et al. [9] provides 3D building models for the Netherlands at multiple levels of detail, including LOD1.2, LOD1.3, and LOD2.2. These 3D models are generated by combining the building footprint data from the BAG with elevation data from the AHN, resulting in a comprehensive and regularly updated dataset suitable for city-scale reconstruction, analysis, and benchmarking of both geometric and semantic aspects of urban environments. The 3D BAG is open-access and is used widely in research applications.

5.2.2 Project PLATEAU

Project PLATEAU dataset from Japan [97] provides a comprehensive 3D city model repository, currently covering over 200 cities, towns, and wards. The dataset is primarily derived from ALS data and is structured according to the CityGML standard, with most buildings represented at LOD1 and LOD2. As an open digital commons, Project PLATEAU supports research, urban planning, and digital twin development by providing consistent, city-scale models that integrate geometric and semantic information across a wide range of urban environments.

5.2.3 GlobalBuildingAtlas

The GlobalBuildingAtlas [98] is a large-scale, openly available dataset providing global coverage of building footprints, heights, and LOD1 3D building models at the individual building level. It is the first open dataset to offer high-quality, consistent, and complete building information in both 2D and 3D formats at a worldwide scale. The dataset comprises approximately 2.68 billion LOD1 building instances, enabling reliable analysis of building geometry and volume across diverse geographic contexts. While limited to LOD1 representations and building entities, it is a valuable reference dataset for city-scale modeling and digital twin applications.

5.2.4 Others

In addition to specific city-scale datasets, recent efforts have focused on compiling and analyzing semantic 3D city model resources at a global level. Notably, Wysocki et al. [21] provides a comprehensive survey of ALS datasets with mesh ground truths, emphasizing semantic 3D city models. Their

work analyzes openly available datasets worldwide, evaluates object-level statistics with distinctions based on object types, and identifies potential datasets suitable for testing 3D reconstruction methods for semantic 3D building models. To further support the community, the authors released a curated repository named *Awesome CityGML*¹, which aggregates references and links to a wide range of semantic 3D city model datasets, serving as a valuable resource for researchers working on urban reconstruction.

Another important resource in this context is the *3D City Database*² (3DCityDB) by Yao et al. [153], an early and widely adopted platform for managing semantic 3D city models. 3DCityDB is a free and open-source geodatabase solution designed for storing, analyzing, and visualizing CityGML-based 3D models. Being a spatial database system, it implements the OGC CityGML 3.0 standard and provides a suite of tools for importing, exporting, visualizing, and integrating 3D city models into automated workflows. The platform enables efficient storage and retrieval of semantic 3D data, making it a practical resource for researchers and practitioners working on urban modeling and reconstruction.

Additional resources from TU Delft further support research in semantic 3D city modeling. An open archive³ of datasets, providing access to various city-scale point clouds and CityGML models for research purposes. Complementing this, Biljecki et al. [99] introduced *Random3DCity*, a procedurally generated synthetic dataset of buildings available at multiple LODs. Although limited to building models, Random3DCity offers a controlled environment for testing reconstruction algorithms and exploring methodological developments without the constraints of real-world data.

6 Findings & Discussion

Since there is no universal definition of city-scale digital twins, semantic 3D city reconstruction, or city-level reconstruction, this scoping review provides a working definition, adapted from the same given by Wysocki et al. [21].

Definition 1 Semantic 3D City Models are defined as urban representations comprising object-level geometry enriched with semantic information, structured within a hierarchical data model that explicitly encodes object-to-object relationships.

These models differ fundamentally from BIM representations, which focus on individual construction sites and

¹ <https://github.com/OloOcki/awesome-citygml>.

² <https://github.com/3dcitydb>.

³ <https://3d.bk.tudelft.nl/opendata/opencities/>.

rely on constructive solid geometry (CSG) as the modeling paradigm.

The scope of this review is narrowed down to the reconstruction of roads and buildings, as these constitute the core structural elements required for any semantic city-level environment. Although comprehensive semantic city models may include vegetation, street furniture, utilities, and other urban entities, the majority of existing large-scale reconstruction research and the papers analyzed in this review primarily address roads and buildings. In this section, the integrated pipeline and the state-of-the-art methods in reconstruction pipelines are discussed.

Generic surface reconstruction techniques, such as Marching Cubes [27], Ball Pivoting Algorithm [32], and Poisson reconstruction [33], continue to be widely used due to their simplicity and efficacy, but they exhibit clear limitations when it comes to the domain of large-scale urban scenes. For instance, Ball Pivoting is intuitive and efficient but depends heavily on uniform sampling density and explicitly assigned radius, making it unreliable for ALS data with occlusions or missing roof/wall regions. On the other hand, Poisson reconstruction, though remarkably robust to noise and capable of producing watertight models, tends to oversmooth sharp urban features, merge narrow gaps, and impose a global regularization that conflicts with the planar, rectilinear nature of man-made structures.

6.1 Integrated Pipeline for 3D Reconstruction

Despite the emergence of automatic DL pipelines, integrated digital twin workflows still face constraints. Models trained on specific datasets are limited due to sensor characteristics or environmental changes. Location-aware segmentation helps detect on-road and roadside entities, but several hybrid systems remain focused purely on road surfaces and markings, without modeling buildings or their geometric relationships [45]. Some methods also introduce high computational costs, especially when computing normals, eigenvalues, contextual features, or Poisson meshes at the city scale. These overheads make it difficult to deploy such pipelines for very large regions without parallel processing or suitable hardware. There is also a scarcity of readily available datasets to support such integrated reconstruction of roads and buildings for semantic 3D city models, as open datasets are not necessarily acquired for this specific purpose. Hence, they fall short on certain characteristics, which may at times be rectifiable through computation. For instance, the DALES dataset [94] has sufficient building information, but the ground points do not have fine-grained differentiation into road points and others. Similarly, many city-scale datasets require additional auxiliary datasets, such as footprint information, for building instantiation. Overall,

there is a gap in publishing ready-to-use datasets to implement the integrated pipeline.

Through this scoping review, we argue that road and building reconstruction should follow a sequence where the reconstructed roads provide a structural prior to building reconstruction. This stems from the requirement of road network topology for building pose estimation and curve modeling of road boundaries. As discussed in Sect. 3.4, building pose estimation is implicitly handled within reconstruction pipelines, where dominant orientations are recovered using geometric methods in the state-of-the-art. However, these orientations can be either predetermined or reinforced by road-derived cues, thus improving alignment and ensuring consistent structural directions across the reconstructed scene. Figure 2 illustrates this sequence through a progressive rendering of a tile in the NPM3D/Paris-Lille dataset, from segmented point clouds through filtered road and building points to the final LOD1-like reconstruction. Here, the road centerlines serve as the structural prior for building orientation and placement.

6.2 Road Reconstruction from LiDAR

This review has focused on ALS, MLS, and TLS. It has been found that most of the methods here rely more on MLS than on ALS or TLS. MLS data offers a strong advantage of dense road points, along with some scans containing navigation data. High-density point clouds enable highly reliable curb detection, edge extraction, and pavement modeling. High-precision at cm-scale for the final road and pavement surface models can be achieved using angular distance to ground normal as a major feature to emphasize vertical structures like curbs [38]. These geometric models adapt smoothly to changing road shapes, intersections, and grade transitions. A few pipelines also integrate auxiliary cues, such as intensity, trajectory information, or topographic maps, which help resolve occlusions, improve continuity, and support accurate centreline and surface estimation [39, 41]. When various inputs are available, an automated pipeline using semantic segmentation, boundary computation, and computing slopes and layout elements in a BIM-ready form can be achieved [44]. Fully automated pipelines are becoming practical as well. For instance, the hybrid MLS method combines various data-driven techniques, trajectory-based partitioning, and Poisson reconstruction to output watertight meshes with semantic labels [45].

Road reconstruction techniques evolved from early geometric pipelines to more recent hybrid and data-driven approaches. Many classical methods focus on geometric reasoning, spline fitting, curb detection, or marking extraction, and these tend to perform well when the underlying road structures are clean, well-marked, and captured with

dense LiDAR. For example, ALS-based techniques, such as modeling complex interchanges with multiple height levels by combining smooth-patch segmentation with plane fitting and 2D map integration [36]. Similarly, spline and parametric curve-based methods [39–41] extract road centerlines and alignment geometry in a lightweight manner, often achieving very high accuracy for highways and long, continuous segments. Some of these pipelines produce interpretable models, use standard curve types, and maintain consistent reconstruction quality when the environment follows planned road layouts.

However, these advantages come with several limitations. A major challenge across many road reconstruction works is the strong dependency on data quality. Methods that rely on curb sharpness or marking clarity quickly degrade in scenes with occlusions, shadows, faded paint, or parked vehicles [38, 39, 41]. ALS-based workflows suffer from low point density and vegetation occlusion in urban areas, while automotive LiDAR lacks the vertical resolution required for complex urban intersections [42, 43]. Also, several methods provide only partial outputs, such as polylines, curbs, spline centerlines, or parametric curves, without reconstructing a continuous road surface [39, 40], and may require additional processing to achieve a full geometric representation.

Generalization remains another concern. Many early works were designed with highways, structured junctions, or clean pavements in mind, and they do not easily extend to dense urban scenes with irregular intersections and unmarked lanes. Classical parametric methods assume good geometry, while active contour or GVF (Gradient Vector Flow) based pipelines [39] are sensitive to point density and parameter tuning. Some frameworks also rely on manual intervention when extracted geometry is noisy or when occlusions break the continuity of the surface [38]. In multi-sensor workflows, calibration requirements, inconsistent sampling patterns, and coordinate alignment often limit portability. Even in more recent BIM-oriented methods [44], where the output is a detailed road entity model with geometric information (alignment, slopes) and infrastructure components in a BIM environment, the final export or modeling stage may depend on proprietary tools, which affects reproducibility.

6.3 Primitive- and Extrusion-Based Building Reconstruction

Primitive-based techniques, such as RANSAC [34], handle noise well and can extract clean geometric elements, but by design, they are limited to predefined primitives and cannot recover complete building shells or complex roof structures. More recent DL models, e.g., NKSR [35] and NKF [154], achieve impressive fidelity on object-level benchmarks,

even though they rely on extensive training, do not generalize to city-scale ALS point clouds, and often struggle with the regularity and topology constraints required for BIMs or CityGML-compliant building models. These limitations collectively motivate domain-specific reconstruction strategies. Urban buildings require planar, watertight, and regularized geometry reconstruction, and they require consistency across thousands of structures, robustness to missing data, and outputs aligned with geospatial standards. Hence, the use of domain-specific techniques is inevitable.

The dependence on predefined geometric assumptions makes building reconstruction automated and scalable. As an example of utilizing predefined shapes, roof-type selection from a library (flat, shed, gabled, hipped, Berliner, and plus junction shapes) was highly generalized and reliable [54]. However, this dependency on predefined primitives also limits their flexibility. Methods that assume planar or rectilinear roof structures struggle with curved, organic, or highly irregular buildings, and performance drops noticeably when the architectural style deviates from the expected template library [52, 54]. Many works also depend heavily on the availability and quality of footprints, cadastral data, or clean roof segmentation. In works such as Vosselman et al. [48] and Awrangjeb and Fraser [59], misaligned footprints, noisy point clouds or incomplete building outlines directly influenced the reconstruction quality and may introduce gaps or distorted geometries. Even in pipelines that include strong footprint regularization, such as alpha shape filtering, Delaunay-driven corner extraction, etc., accuracy varies depending on density, vegetation, and the amount of noise in the data [49, 51, 60].

One key limitation of primitive- and extrusion-based methods is the sensitivity to preprocessing. Many pipelines depend on accurate ground separation, roof segmentation, or planarity checks, and errors in these early steps propagate into the final model, especially in cluttered areas or lower-density ALS scans [51, 58]. Also, many of the methods focus on certain areas, such as the roof topology, and not the full watertight building volume, requiring external steps for façade completion or vertical extrusion [59, 61], since reliable higher LODs reconstruction using only primitive/extrusion/geometric methods is rare. Moreover, several approaches are tailored specifically for ALS data and do not readily handle MLS, TLS, or point clouds with additional features, making generalization across sensors less straightforward.

For multi-modal data, including LiDAR, relying on DSM data tends to lose sharp geometric features, and interpolation artifacts may distort roof edges or small structures if the DSM resolution is poor [52]. Similarly, the use of aerial imagery for texture mapping improves visual quality, but it

also brings issues related to occlusion, illumination inconsistencies, and missing façade information [10].

In general, primitive or extrusion-based reconstruction provides a consistent, computationally efficient solution for urban environments with structured geometry. It provides interpretable models, strong scalability, and clean footprints, but it remains constrained by rigid shape assumptions, preprocessing dependencies, and limited adaptability to complex or unconventional architectures [25]. These limitations become more evident when dealing with curved roofs, free-form buildings, dense vegetation, missing footprints, or noisy point clouds, where primitive-based fitting either breaks down or produces oversimplified models. In addition, most of these methods rely on early segmentation and planarity checks, meaning that errors introduced in the initial steps will propagate directly to the final reconstruction.

6.4 Data-Driven and Polygonal Surface Reconstruction for Buildings

Because of these challenges, many recent works have moved toward data-driven or polygonal surface reconstruction methods that do not rely on predefined primitive libraries and can adapt to a wider variety of building forms. This shift also aligns with the increasing availability of large datasets and improved GPU efficiency, enabling end-to-end pipelines that operate directly on raw point clouds without requiring strict geometric constraints. In practice, data-driven and polygonal methods overcome several limitations of primitive pipelines by providing better reconstruction of complex topologies, improved handling of missing data, and the ability to incorporate multimodal inputs when available. As a result, they have become the preferred choice for scenarios where flexibility, detail preservation, and generalization across diverse building styles are more important than strict runtime efficiency.

Several methods utilizing data-driven or polygonal surface reconstruction approaches provide fully automatic reconstruction from raw point clouds, reducing manual intervention and increasing scalability across large urban areas [66, 71, 74]. In fact, some methods employing modern architectures, such as reinforcement learning and transformer-based models, allow direct encoding of building points and mesh generation without requiring predefined geometric primitives, thereby supporting arbitrarily shaped roofs while preserving wall verticality [65, 70]. The use of transformers or sparse CNNs also demonstrates strong generalization across different datasets [70, 73].

Irrespective of the reconstruction strategy, many of the methods discussed require either annotated or pre-segmented datasets, or rely on a separate preprocessing pipeline to obtain such segmentation [71]. Therefore, even though

the manual intervention required is significantly reduced in advanced data-driven approaches [73] or complex polygonal surface reconstruction frameworks [64], the considerably higher computational cost becomes a disadvantage when compared to primitive-based methods.

However, despite this computational overhead, the ability to operate at large scales while achieving high geometric accuracy and detailed roof preservation remains a major advantage for several methods [64, 66]. Data-driven models such as [71], which incorporate GAN-based completion strategies, also handle missing or incomplete data very effectively, yielding sharper edges and more complete reconstructed surfaces.

Another notable advantage of these advanced techniques is the capacity to incorporate multimodal data [55, 67, 68]. This semantic enrichment through multimodal fusion enables applications such as heritage-focused digital twins [69, 155] and CityGML-compliant models [68]. However, these methods are often used in scene- or domain-specific contexts, which is unsurprising given the difficulty of acquiring rich multimodal data at a city scale. Consequently, large-scale adaptation of such models might fail due to limited generalization. These approaches are, therefore, more suitable when the primary objective is detailed, domain-specific reconstruction.

7 Conclusion

Two key takeaways from our scoping review are that LiDAR scans are critical for 3D geometric reconstruction for digital twin applications, and that roads and buildings are objects that can be prioritized in a specific order for 3D reconstruction. There is a current need to take stock of the gaps, given the technological advances in large-scale data acquisition, the artificial intelligence ecosystem, and the need for automation in urban planning from both technology and policy points of view. The short-term future work entails curating datasets for improving learning models in 3D reconstruction of roads and buildings, and developing appropriate workflows for diverse urban settlements and environments. The long-term future work entails improving information content in the data using multimodal datasets and designing an automated integrated pipeline for scalable, secure, and privacy-preserving reconstruction.

As a concluding remark, this scoping review has focused on the usability of LiDAR point clouds for semantic 3D city model generation. Restricting the 3D city models to roads and buildings as their structural elements, this study reviewed the state-of-the-art on their individual reconstruction pipelines holistically. This review, in that regard, is one of a kind on an integrated reconstruction pipeline.

Acknowledgements International Institute of Information Technology Bangalore, and the Mphasis Center for Cognitive Computing (for K.N.'s Master's fellowship); Blekinge Institute of Technology, Sweden (for open access).

Author contribution K. N. performed exhaustive research, curating information, image generation, and manuscript writing; J.S.-N. contributed to the conceptualization, writing, and proofreading; P.G. contributed to review planning, writing, and proofreading. All authors read and approved the final manuscript.

Funding Open access funding provided by Blekinge Institute of Technology.

Data Availability No datasets were generated or analysed during the current study.

Materials Availability Not applicable.

Code Availability Not applicable.

Declarations

Ethics Approval and Consent to Participate Not applicable.

Consent for Publication Not applicable.

Use of Generative AI and AI-Assisted Technologies Large language Models were used solely for language refinement and proofreading.

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Wang R, Peethambaran J, Chen D (2018) LiDAR point clouds to 3-D urban models: a review. *IEEE J Sel Topics Appl Earth Observ Remote Sens* 11(2):606–627. <https://doi.org/10.1109/JSTARS.2017.2781132>
- Ma Z, Liu S (2018) A review of 3D reconstruction techniques in civil engineering and their applications. *Adv Eng Inf* 37:163–174. <https://doi.org/10.1016/j.aei.2018.05.005>
- Groger G, Kolbe TH, Czerwinski A, Nagel C (2008) OpenGIS® city geography markup language (CityGML) encoding standard. Version 1.0.0. Open Geospatial Consortium. <https://doi.org/10.25607/OBP-635>
- Löwner M-O, Joachim B, Gröger G, U. G, Häfele K-H, Schülter S (2012). CityGML 2.0—an International standard for 3D city models. Part 1: data model
- Kutzner T, Smyth C, Nagel C, Coors V, Vinasco-Alvarez D, Ishimaru N, Yao Z, Heazel C, Kolbe TH (2023). OGC city geography markup language (CityGML) version 3.0 Part 2: GML encoding standard
- Ledoux H, Arroyo Otori 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, Softw Stand* 4(1):1–12. <https://doi.org/10.1186/s40965-019-0064-0>
- Biljecki F (2020) Exploration of open data in Southeast Asia to generate 3D building models. *ISPRS annals of Photogrammetry, Remote sensing and spatial information Sciences. ISPRS Ann Photogramm Remote Sens Spatial Inf Sci VI-4/W1-2020:37–44*. <https://doi.org/10.5194/isprs-annals-vi-4-w1-2020-37-2020>
- Döllner J, Kolbe TH, Liecke F, Sgouros T, Teichmann K (2006) The virtual 3D city model of Berlin - managing, integrating, and communicating complex urban information. In *Proceedings of the 25th Urban Data Management Symposium UDMS*
- Peters R, Dukai B, Vitalis S, Liempt J, Stoter J (2022) Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. *Photogramm Eng Remote Sens* 88(3):165–170. <https://doi.org/10.14358/PERS.21-00032R2>
- Poullis C, You S (2009) Photorealistic large-scale urban city model reconstruction. *IEEE Trans Visual Comput Graphics* 15(4):654–669. <https://doi.org/10.1109/TVCG.2008.189>
- Xu Y, Stilla U (2021) Toward building and civil infrastructure reconstruction from point clouds: a review on data and key techniques. *IEEE J Sel Topics Appl Earth Observ Remote Sens* 14:2857–2885. <https://doi.org/10.1109/JSTARS.2021.3060568>
- Roynard X, Deschaud J-E, Goulette F (2018) Paris-Lille-3D: a large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Int J Robot Res* 37(6):545–557. <https://doi.org/10.1177/0278364918767506>
- Yu B, Wang Y, Chen Q, Chen X, Zhang Y, Luan K, Ren X (2024) A review of road 3D modeling based on Light detection and Ranging point clouds. *J Educ Chang Road Eng* 4(4):386–398. <https://doi.org/10.1016/j.jreng.2024.04.009>
- Ledoux H, Biljecki F, Dukai B, Kumar K, Peters R, Stoter J, Commandeur T (2021) 3dfier: automatic reconstruction of 3D city models. *J Retailing Open Source Softw* 6(57):2866. <https://doi.org/10.21105/joss.02866>
- Huang J, Stoter J, Peters R, Nan L (2022) City3D: large-scale building reconstruction from airborne LiDAR point clouds. *Remote Sens* 14(9):2254. <https://doi.org/10.3390/rs14092254>
- Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, Savarese S, Savva M, Song S, Su H et al. (2015 arXiv preprint arXiv:1512.03012) ShapeNet: an information-rich 3D model repository. <https://doi.org/10.48550/arXiv.1512.03012>
- Sun J, Zhang Q, Kailkhura B, Yu Z, Xiao C, Mao ZM (2022) ModelNet40-C: a robustness benchmark for 3D point cloud Recognition under corruption. In *ICLR 2022 Workshop on Socially Responsible Machine Learning*, vol 7
- Buyukdemircioglu M, Kocaman S, Kada M (2022) Deep learning for 3D building reconstruction: a review. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLIII-B2-2022:359–366*. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-359-2022>
- Abreu N, Pinto A, Matos A, Pires M (2023) Procedural point cloud modelling in scan-to-BIM and Scan-vs-BIM applications: a review. *ISPRS Int J Geo-Inf* 12(7):260. <https://doi.org/10.3390/ijgi12070260>
- Huang Z, Wen Y, Wang Z, Ren J, Jia K (2024) Surface reconstruction from point clouds: a survey and a benchmark. *IEEE Trans Pattern Anal Mach Intell* 46(12):9727–9748. <https://doi.org/10.1109/TPAMI.2024.3429209>

21. Wysocki O, Schwab B, Beil C, Holst C, Kolbe TH (2024) Reviewing open data semantic 3D city models to develop novel 3D reconstruction methods. *Int Arch Photogramm Remote Sens Spatial Inf. Sci XLVIII-4-2024*:493–500. <https://doi.org/10.5194/isprs-archives-XLVIII-4-2024-493-2024>
22. Aydin M (2025) Analyzing the impact of ISO 16739-1: 2024 (Industry foundation classes, IFC) on data sharing and building information modeling (BIM) collaboration in the construction Industry. *J Architectural Sci Appl* 10(1):157–174. <https://doi.org/10.30785/mbud.1609588>
23. Hu Z-Z, Leng S, Lin J-R, Li S-W, Xiao Y-Q (2022) Knowledge extraction and discovery based on BIM: a critical review and future directions. *Arch Computat Methods Eng* 29(1):335–356. <https://doi.org/10.1007/s11831-021-09576-9>
24. Tarsha Kurdi F, Lewandowicz E, Gharineiat Z, Shan J (2025) High-resolution building indicator mapping using airborne LiDAR data. *Electronics* 14(9):1821. <https://doi.org/10.3390/electronics14091821>
25. Chen D, Zhu C, Zhang Z, Na J, Shen Y, Chen Y, Peethambaran J, Zhang L (2025) Public building geometric models from point clouds: a multidimensional quality evaluation framework. *IEEE Geosci Remote Sens Mag* 2–29. <https://doi.org/10.1109/MGRS.2025.3618468>
26. Biljecki F, Ledoux H, Stoter J (2016) An improved LoD specification for 3D building models. *Comput, Environ Urban Syst* 59:25–37. <https://doi.org/10.1016/j.compenvurbysys.2016.04.005>
27. Lorensen WE, Cline HE (1987) Marching Cubes: a high resolution 3D surface construction algorithm. *SIGGRAPH Comput Graph* 21(4):163–169. <https://doi.org/10.1145/37402.37422>
28. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1992) Surface reconstruction from unorganized points. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, pp 71–78. <https://doi.org/10.1145/133994.1340>
29. Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, Evans TR (2001) Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH)*, pp 67–76. <https://doi.org/10.1145/383259.383266>
30. Berger M, Tagliasacchi A, Seversky LM, Alliez P, Guennebaud G, Levine JA, Sharf A, Silva CT (2017) A survey of surface reconstruction from point clouds. In: *Computer graphics forum*, vol 36. Wiley Online Library, pp 301–329. <https://doi.org/10.1111/cgf.12802>
31. Liao Y, Donne S, Geiger A (2018) Deep Marching Cubes: learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 2916–2925. <https://doi.org/10.1109/CVPR.2018.00308>
32. Bernardini F, Mittleman J, Rushmeier H, Silva C, Taubin G (1999) The ball-Pivoting algorithm for surface reconstruction. *IEEE Trans Visual Comput Graphics* 5(4):349–359. <https://doi.org/10.1109/2945.817351>
33. Kazhdan M, Bolitho M, Hoppe H (2006) Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, vol 7. <https://doi.org/10.2312/SGP/SGP06/061-070>
34. Schnabel R, Wahl R, Klein R (2007) Efficient RANSAC for point-cloud shape detection. *Comput Graphics Forum* 26(2):214–226. <https://doi.org/10.1111/j.1467-8659.2007.01016.x>. Wiley Online Library
35. Huang J, Gojcic Z, Atzmon M, Litany O, Fidler S, Williams F (2023) Neural kernel surface reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 4369–4379. <https://doi.org/10.1109/CVPR52729.2023.00425>
36. Oude Elberink SJ, Vosselman G (2009) 3D information extraction from laser point clouds covering complex road junctions. *Photogramm Rec* 24(125):23–36. <https://doi.org/10.1111/j.1477-9730.2008.00516.x>
37. McElhinney CP, Kumar P, Cahalane C, McCarthy T (2010). Initial results from European road safety inspection (eursi) mobile mapping project. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVIII, Part 5, Commission V Symposium, Newcastle, 440–445
38. Hervieu A, Soheilian B (2013) Semi-automatic road/pavement modeling using mobile laser scanning. *ISPRS annals of the Photogrammetry, Remote sensing and spatial information Sciences*. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci* II-3/W3:31–36. <https://doi.org/10.5194/isprsannals-II-3-W3-31-2013>
39. Kumar P, McElhinney CP, Lewis P, McCarthy T (2013) An automated algorithm for extracting road edges from Terrestrial mobile LiDAR data. *ISPRS J Photogramm* 85:44–55. <https://doi.org/10.1016/j.isprsjprs.2013.08.003>
40. Garach L, Oña J, Pasadas M (2014) Mathematical formulation and preliminary testing of a spline approximation algorithm for the extraction of road alignments. *Automation Construct* 47:1–9. <https://doi.org/10.1016/j.autcon.2014.07.002>
41. Zhang Z, Li J, Guo Y, Yang C, Wang C (2020) 3D highway curve reconstruction from mobile laser Scanning point clouds. *IEEE Trans Intell Transp Syst* 21(11):4762–4772. <https://doi.org/10.1109/TITS.2019.2946259>
42. Katkoria D, Sreevalsan-Nair J (2022) RoSELS: road surface extraction for 3D automotive LiDAR PointCloud sequence. In *Proceedings of the 3rd International Conference on Deep Learning Theory and Applications (DeLTA)*, pp 55–67. <https://doi.org/10.5220/0011301700003277>
43. Katkoria D, Sreevalsan-Nair J (2023) Evaluating and improving RoSELS for road surface extraction from 3D automotive LiDAR point cloud sequences. In: *Deep learning theory and applications: third international conference, DeLTA 2022, Lisbon, Portugal, July 12–14, 2022, revised selected papers*. Springer Nature, p 98. https://doi.org/10.1007/978-3-031-37317-6_6
44. Wang Y, Wang W, Liu J, Chen T, Wang S, Yu B, Qin X (2023) Framework for geometric information extraction and digital modeling from LiDAR data of road scenarios. *Remote Sens* 15(3):576. <https://doi.org/10.3390/rs15030576>
45. Davletshina D, Reja VK, Brilakis I (2024) Automating construction of road digital twin geometry using context and location aware segmentation. *Automation Construct* 168:105795. <https://doi.org/10.1016/j.autcon.2024.105795>
46. Liao Y, Xie J, Geiger A (2023) KITTI-360: a novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Trans Pattern Anal Mach Intell* 45(3):3292–3310. <https://doi.org/10.1109/TPAMI.2022.3179507>
47. Vajdic N (2025) Digital roads of the future Annual review 2024. <https://www.repository.cam.ac.uk/bitstreams/2d54bae8-acd3-4c85-91f9-4ddd9e086294/download>. Last accessed on January 20, 2026
48. Vosselman G, Dijkman S et al. (2001) 3D building model reconstruction from point clouds and ground plans. *Int Archives Photogrammetry Remote Sens Spatial Inf Sci* 34(3/W4):37–44
49. Zhang K, Yan J, Chen S-C (2006) Automatic construction of building footprints from airborne LIDAR data. *IEEE Trans. Geosci. Remote Sens* 44(9):2523–2533. <https://doi.org/10.1109/TGRS.2006.874137>
50. Poullis C, You S, Neumann U (2008) Rapid creation of large-scale Photorealistic virtual environments. In *2008 IEEE Virtual Reality Conference*, IEEE, pp. 153–160. <https://doi.org/10.1109/VR.2008.4480767>

51. Dorninger P, Pfeifer N (2008) A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser Scanning point clouds. *Sensors* 8(11):7323–7343. <https://doi.org/10.3390/s8117323>
52. Lafarge F, Descombes X, Zerubia J, Pierrot-Deseilligny M (2008) Structural approach for building reconstruction from a single DSM. *IEEE T Pattern Anal* 32(1):135–147. <https://doi.org/10.1109/TPAMI.2008.281>
53. Zhou Q-Y, Neumann U. (2009) U.: a streaming framework for seamless building reconstruction from large-scale aerial LiDAR data. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp. 2759–2766. <https://doi.org/10.1109/CVPR.2009.5206760>
54. Kada M, McKinley L (2009) 3D building reconstruction from LiDAR based on a cell decomposition approach. *Int Archives Photogrammetry, Remote Sens Spatial Inf Sci* 38(Part 3):4
55. Poullis C, You S (2009) Automatic reconstruction of cities from remote sensor data. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp. 2775–2782. <https://doi.org/10.1109/CVPR.2009.5206562>
56. Vanegas CA, Aliaga DG, Benes B (2012) Automatic extraction of Manhattan-world building masses from 3D laser range scans. *IEEE Trans Visual Comput Graphics* 18(10):1627–1637. <https://doi.org/10.1109/TVCG.2012.30>
57. Zhou Q-Y, Neumann U (2012) 2.5D building modeling by discovering global regularities. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp. 326–333. <https://doi.org/10.1109/CVPR.2012.6247692>
58. Chen D, Zhang L, Mathiopoulos PT, Huang X (2014) A methodology for automated segmentation and reconstruction of urban 3-D buildings from ALS point clouds. *IEEE J Sel Top Appl Earth Observations Remote Sens* 7(10):4199–4217. <https://doi.org/10.1109/JSTARS.2014.2349003>
59. Awrangjeb M, Fraser CS (2014) Automatic segmentation of raw LIDAR data for extraction of building roofs. *Remote Sens* 6(5):3716–3751. <https://doi.org/10.3390/rs6053716>
60. Awrangjeb M (2016) Using point cloud data to identify, trace, and regularize the outlines of buildings. *Int J Remote Sens* 37(3):551–579. <https://doi.org/10.1080/01431161.2015.1131868>
61. Chen D, Wang R, Peethambaran J (2017) Topologically aware building rooftop reconstruction from airborne laser Scanning point clouds. *IEEE T Geosci Remote* 55(12):7032–7052. <https://doi.org/10.1109/TGRS.2017.2738439>
62. Zhang W, Li Z, Shan J (2021) Optimal model fitting for building reconstruction from point clouds. *IEEE J Sel Topics Appl Earth Observ Remote Sens* 14:9636–9650. <https://doi.org/10.1109/JSTARS.2021.3110429>
63. Lewandowicz E, Tarsha Kurdi F, Gharineiat Z (2022) 3D LoD2 and LoD3 modeling of buildings with ornamental Towers and turrets based on LiDAR data. *Remote Sens* 14(19):4687. <https://doi.org/10.3390/rs14194687>
64. Nan L, Wonka P (2017) PolyFit: polygonal surface reconstruction from point clouds. In Proceedings of the IEEE International Conference on Computer Vision, pp 2353–2361. <https://doi.org/10.1109/ICCV.2017.258>
65. Zhang L, Zhang L (2017) Deep learning-based classification and reconstruction of residential scenes from large-scale point clouds. *IEEE T Geosci Remote* 56(4):1887–1897. <https://doi.org/10.1109/TGRS.2017.2769120>
66. Zhang L, Li Z, Li A, Liu F (2018) Large-scale urban point cloud labeling and reconstruction. *ISPRS J Photogramm* 138:86–100. <https://doi.org/10.1016/j.isprsjprs.2018.02.008>
67. Chen J, Kira Z, Cho YK (2019) Deep learning approach to point cloud scene understanding for automated scan to 3D reconstruction. *J Comput Civ Eng* 33(4):04019027. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000842](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000842)
68. Wysocki O, Schwab B, Hoegner L, Kolbe TH, Stilla U (2021) Plastic surgery for 3D city models: a pipeline for automatic geometry refinement and semantic enrichment. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci* 4:17–24. <https://doi.org/10.5194/isprs-annals-V-4-2021-17-2021>
69. Pan X, Lin Q, Ye S, Li L, Guo L, Harmon B (2024) Deep learning based approaches from semantic point clouds to semantic BIM models for heritage digital twin. *Herit Sci* 12(1):65. <https://doi.org/10.1186/s40494-024-01179-4>
70. Liu Y, Obukhov A, Wegner JD, Schindler K (2024) Point2Building: reconstructing buildings from airborne LiDAR point clouds. *ISPRS J Photogramm* 215:351–368. <https://doi.org/10.1016/j.isprsjprs.2024.07.012>
71. Ding Z, Lu Y, Shao S, Qin Y, Lu M, Song Z, Sun D (2025) Research on 3D reconstruction methods for incomplete building point clouds using deep learning and geometric primitives. *Remote Sens* 17(3):399. <https://doi.org/10.3390/rs17030399>
72. Xiao X, Wang K, Zhong Z, Qu W, Wu W, Cui Z, Su Y, Li A, Gong J, Li D (2025) A novel data-driven based high-precision building roof contour full-automatic extraction and structured 3D reconstruction method combining stereo images and LiDAR points. *Int J Digit Earth* 18(1):2484668. <https://doi.org/10.1080/17538947.2025.2484668>
73. Liu Y, Wang R, Huang S, Cai G (2025) EdgeDiff: edge-aware diffusion network for building reconstruction from point clouds. In 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 17008–17018. <https://doi.org/10.1109/CVPR52734.2025.01585>
74. Bui DM, Kim H, Youn J, Kim C (2025) Automated 3D building model reconstruction using orthophotos and point clouds. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLVIII-G-2025-233–239*. <https://doi.org/10.5194/isprs-archives-XLVIII-G-2025-233-2025>
75. Li J, Lv R, Lan Q, Shou X, Ruan H, Cao J, Li Z (2025) Deep line-segment detection- driven building footprints extraction from backpack LiDAR point clouds for urban scene reconstruction. *Remote Sens* 17(22):3730. <https://doi.org/10.3390/rs17223730>
76. Haklay M, Weber P (2008) OpenStreetMap: user-generated street maps. *IEEE Pervasive Comput* 7(4):12–18. <https://doi.org/10.1109/MPRV.2008.80>
77. Sreevalsan-Nair J, Jindal A, Kumari B (2018) Contour extraction in buildings in airborne LiDAR point clouds using multiscale local geometric descriptors and Visual analytics. *IEEE J Sel Top Appl Earth Observations Remote Sens* 11(7):2320–2335. <https://doi.org/10.1109/JSTARS.2018.2833801>. IEEE
78. ASAM e.V. (2020) 1.6 Base Standard. Technical report. ASAM OpenDRIVE - Road Description format. https://releases.asam.net/OpenDRIVE/1.6.0/ASAM_OpenDRIVE_BS_V1-6-0.html (29 March 2020)
79. Hug C, Krzystek P, Fuchs W (2004) Advanced LiDAR data processing with LasTools. XXth ISPRS Congr 12–23
80. Kanopoulos N, Vasanthavada N, Baker RL (1988) Design of an image edge detection filter using the Sobel operator. *IEEE J Solid-State Circuits* 23(2):358–367. <https://doi.org/10.1109/4.996>
81. Illingworth J, Kittler, Kittler J (1988) J.: a survey of the Hough transform. *Computer Vision, Graphics, and image processing. Comput Vision, Graphics, Image Process* 44(1):87–116. [https://doi.org/10.1016/S0734-189X\(88\)80033-1](https://doi.org/10.1016/S0734-189X(88)80033-1)
82. Niemeyer J, Rottensteiner F, Soergel U (2014) Contextual classification of LiDAR data and building object detection in urban areas. *ISPRS J Photogramm* 87:152–165. <https://doi.org/10.1016/j.isprsjprs.2013.11.001>
83. Hackel T, Savinov N, Ladicky L, Wegner JD, Schindler K, Pollefeys M (2017). Semantic3D. net: a New large-scale point cloud classification benchmark. *ISPRS Annals of the Photogrammetry,*

- Remote Sensing and Spatial Information Sciences IV-1/W1, 91–98. <https://doi.org/10.5194/isprs-annals-IV-1-W1-91-2017>
84. Munoz D, Bagnell JA, Vandapel N, Hebert M (2009) Contextual classification with functional Max-Margin Markov Networks. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, 975–982. <https://doi.org/10.1109/CVPR.2009.5206590>
 85. Tan W, Qin N, Ma L, Li Y, Du J, Cai G, Yang K, Li J (2020) Toronto-3D: a large-scale mobile LiDAR dataset for semantic segmentation of urban roadways. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp 202–203. <https://doi.org/10.1109/CVPRW50498.2020.00109>
 86. Armeni I, Sener O, Zamir AR, Jiang H, Brilakis I, Fischer M, Savarese S (2016) 3D semantic parsing of large-scale indoor spaces. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1534–1543. <https://doi.org/10.1109/CVPR.2016.170>
 87. Cesium. (2025). The platform for 3D geospatial. <https://cesium.com/>. Accessed 25 Jan
 88. Wang R, Huang S, Yang H (2023) Building3D: an urban-scale dataset and benchmarks for learning roof structures from point clouds. In 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp 20019–20029. <https://doi.org/10.1109/ICCV51070.2023.01837>
 89. Gioi R, Jakubowicz J, Morel J-M, Randall G (2010) LSD: a fast line segment detector with a false detection control. *IEEE T Pattern Anal* 32(4):722–732. <https://doi.org/10.1109/TPAMI.2008.300>
 90. Weinmann M, Jutzi B, Mallet C (2013) Feature relevance assessment for the semantic interpretation of 3D point cloud data. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci II-5/W2:313–318*. <https://doi.org/10.5194/isprsannals-II-5-W2-313-2013>
 91. Weinmann M, Jutzi B, Hinz S, Mallet C (2015) Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J Photogramm* 105:286–304. <https://doi.org/10.1016/j.isprsjprs.2015.01.016>
 92. Gehring J, Hebel M, Arens M, Stilla U (2017) An approach to extract moving objects from MLS data using a volumetric background representation. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci IV-1/W1:107–114*. <https://doi.org/10.5194/isprs-annals-IV-1-W1-107-2017>
 93. Zhu J, Gehring J, Huang R, Borgmann B, Sun Z, Hoegner L, Hebel M, Xu Y, Stilla U (2020) TUM-MLS-2016: an annotated mobile LiDAR dataset of the TUM city Campus for semantic point cloud interpretation in urban areas. *Remote Sens* 12(11):1875. <https://doi.org/10.3390/rs12111875>
 94. Varney N, Asari VK, Graehling Q (2020) DALES: a large-scale aerial LiDAR data set for semantic segmentation. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 717–726. <https://doi.org/10.1109/CVPRW50498.2020.00101>
 95. Kölle M, Laupheimer D, Schmohl S, Haala N, Rottensteiner F, Wegner JD, Ledoux H (2021) The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from UAV LiDAR and Multi-View-Stereo. *ISPRS Open J Photogramm Remote Sens* 1:100001. <https://doi.org/10.1016/j.ophoto.2021.100001>
 96. Nederland AH (2018) AHN3. Digitale Hoogtekaart. <https://ahn.aeregionline.nl/ahnviewer>
 97. Seto T, Furuhashi T, Uchiyama Y (2023) Role of 3D city model data as open digital commons: a case study of openness in Japan's digital twin "PROJECT PLATEAU". *The Int Archives The Photogrammetry, Remote Sens Spatial Inf Sci* 48:201–208. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W7-2023-201-2023>
 98. Zhu XX, Chen S, Zhang F, Shi Y, Wang Y (2025) GlobalBuildingAtlas: an open global and complete dataset of building polygons, heights and LoD1 3D models. *Earth Syst. Sci. Data* 17(12):6647–6668. <https://doi.org/10.5194/essd-17-6647-2025>
 99. Biljecki F, Ledoux H, Stoter J (2016). Generation of multi-LOD 3D city models in CityGML with the Procedural modelling engine Random3Dcity. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4/W1*, 51–59. <https://doi.org/10.5194/isprs-annals-IV-4-W1-51-2016>
 100. Cheng L, Chen S, Liu X, Xu H, Wu Y, Li M, Chen Y (2018) Registration of laser scanning point clouds: a review. *Sensors* 18(5):1641. <https://doi.org/10.3390/s18051641>
 101. Huang X, Mei G, Zhang J, Abbas (2021) R.: a comprehensive survey on point cloud registration. *CoRr abs/2103.02690* <https://doi.org/10.48550/arXiv.2103.02690>
 102. Ning X, Li F, Tian G, Wang Y (2018) An efficient outlier removal method for scattered point cloud data. *PLoS One* 13(8):1–22. <https://doi.org/10.1371/journal.pone.0201280>
 103. Weyrich T, Pauly M, Keiser R, Heinze S, Scandella S, Gross M (2004) Post-processing of scanned 3D surface data. In Proceedings of the First Eurographics Conference on Point-Based Graphics. SPBG'04, pp 85–94
 104. Keller P, Kreylos O, Vanco M, Hering-Bertram M, Cowgill ES, Kellogg LH, Hamann B, Hagen H (2010) Extracting and visualizing structural features in environmental point cloud LiDAR data sets. In: Pascucci V, Tricoche X, Hagen H, Tierny J (eds) *Topological methods in Data analysis and visualization: theory, algorithms, and applications*. pp 179–192. https://doi.org/10.1007/978-3-642-15014-2_15
 105. Carrilho A, Galo M, Santos R (2018) Statistical outlier detection method for airborne LiDAR data. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLII-1:87–92*. <https://doi.org/10.5194/isprs-archives-XLII-1-87-2018>
 106. Zhang Z (1994) Iterative point matching for registration of free-form curves and surfaces. *Int J Comput Vision* 13(2):119–152. <https://doi.org/10.1007/BF01427149>
 107. Rusu RB, Blodow N, Marton Z, Soos A, Beetz M (2007) Towards 3D object maps for autonomous household robots. In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 3191–3198. <https://doi.org/10.1109/IROS.2007.4399309>
 108. Balta H, Velagic J, Bosschaerts W, De Cubber G, Siciliano B (2018) Fast statistical outlier removal based method for large 3D point clouds of outdoor environments. *IFAC-Papersonline* 51(22):348–353. <https://doi.org/10.1016/j.ifacol.2018.11.566>. 12th IFAC Symposium on Robot Control SYROCO 2018
 109. Duan Y, Yang C, Chen H, Yan W, Li H (2021) Low-complexity point cloud denoising for LiDAR by PCA-based dimension reduction. *Optics Commun* 482:126567. <https://doi.org/10.1016/j.optcom.2020.126567>
 110. Cheng D, Zhao D, Zhang J, Wei C, Tian D (2021) PCA-based denoising algorithm for outdoor LiDAR point cloud data. *Sensors* 21(11):3703. <https://doi.org/10.3390/s21113703>
 111. Zou B, Qiu H, Lu Y (2020) Point cloud reduction and denoising based on optimized downsampling and bilateral filtering. *IEEE Access* 8:136316–136326. <https://doi.org/10.1109/ACCESS.2020.3011989>
 112. Guoqiang W, Hongxia Z, Zhiwei G, Wei S, Dagong J (2023) Bilateral filter denoising of LiDAR point cloud data in automatic driving scene. *Infrared Phys Technol* 131:104724. <https://doi.org/10.1016/j.infrared.2023.104724>
 113. Hermosilla P, Ritschel T, Ropinski T (2019) Total denoising: unsupervised learning of 3D point cloud cleaning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 52–60. <https://doi.org/10.1109/ICCV.2019.00014>
 114. Tzermia C, Malamos AG (2025) Comparative analysis of downsampling techniques for machine learning on cultural heritage objects. In Proceedings of the 30th International Conference on

- 3D Web Technology, pp 1–10. <https://doi.org/10.1145/3746237.3746301>
115. Moenning C, Dodgson, NA (2003) Fast Marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory. <https://doi.org/10.48456/tr-562>
 116. Rusinkiewicz S, Levoy M (2001) Efficient variants of the ICP algorithm. In Proceedings Third International Conference on 3-D Digital Imaging and Modeling, pp 145–152. <https://doi.org/10.1109/IM.2001.924423>. IEEE
 117. Lyu W, Ke W, Sheng H, Ma X, Zhang H (2024) Dynamic downsampling algorithm for 3D point cloud map based on voxel filtering. *Appl Sci* 14(8):3160. <https://doi.org/10.3390/app14083160>
 118. Comino-Trinidad M, Chica A et al. (2023). Revisiting Poisson-disk subsampling for massive point cloud decimation. *CoRR* abs/2311.17604 (<https://doi.org/10.48550/arXiv.2311.17604>)
 119. Goswami P, Erol F, Mukhi R, Pajarola R, Gobbetti E (2013) An efficient multi-resolution framework for high quality interactive rendering of massive point clouds using multi-way kd-trees. *Vis Comput* 29(1):69–83. <https://doi.org/10.1007/s00371-012-0675-2>
 120. Goswami P, Zhang Y, Pajarola R, Gobbetti E (2010) High quality Interactive rendering of massive point models using multi-way KD-trees. In 2010 18th Pacific Conference on Computer Graphics and Applications, IEEE, pp. 93–100. <https://doi.org/10.1109/PacificGraphics.2010.20>
 121. Tang Q, Zhang L, Lan G, Shi X, Duanmu X, Chen K (2023) A classification method of point clouds of transmission line corridor based on improved random forest and multi-scale features. *Sensors* 23(3):1320. <https://doi.org/10.3390/s23031320>
 122. Demantké J, Vallet B, Paparoditis N (2012) Streamed vertical rectangle detection in Terrestrial laser scans for facade database production. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci* 1-3:99–104. <https://doi.org/10.5194/isprsannals-1-3-99-2012>
 123. Girardeau-Montaut D et al. (2025) CloudCompare, 3D point cloud and mesh processing software (version 2.13.1). Last accessed on December 01, <https://www.cloudcompare.org/>
 124. Demantké J, Mallet C, David N, Vallet B (2011). Dimensionality based scale selection in 3D LiDAR point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII-5/W12*, 97–102. <https://doi.org/10.5194/isprsarchives-XXXVIII-5-W12-97-2011>
 125. Chajaei F, Bagheri H (2025) LOD1 3D city model from LiDAR: the impact of segmentation accuracy on quality of urban 3D modeling and morphology extraction. *Remote Sens Appl: Soc Environ* 38:101534. <https://doi.org/10.1016/j.rsase.2025.101534>
 126. Li J, Li Y, Chen Y, Fan H, Wang R (2026) City-scale building instance segmentation from LiDAR point clouds via structure-aware method. *Int J Appl Earth Observ Geoinf* 146:105086. <https://doi.org/10.1016/j.jag.2026.105086>
 127. Sreevalsan-Nair J, Kumari B (2017) Local geometric descriptors for multi-scale probabilistic point classification of airborne LiDAR point clouds. *Modeling, Anal, Visual Anisotropy* 175–200. https://doi.org/10.1007/978-3-319-61358-1_8
 128. Vinodkumar PK, Karabulut D, Avots E, Ozcinar C, Anbarjafari (2023) G.: a survey on deep learning based segmentation, detection and classification for 3D point clouds. *Entropy* 25(4):635. <https://doi.org/10.3390/e25040635>
 129. Li X, Chen (2022) D.: a survey on Deep learning-based panoptic segmentation. *Digit Signal Process* 120:103283. <https://doi.org/10.1016/j.dsp.2021.103283>
 130. Dey EK, Awrangjeb M, Tarsha Kurdi F, Stantic B (2023) Machine learning-based segmentation of aerial LiDAR point cloud data on building roof. *Eur J Remote Sens* 56(1):2210745. <https://doi.org/10.1080/22797254.2023.2210745>
 131. Ester M, Kriegel H-P, Sander J, Xu X et al. (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD'96, pp 226–231. <https://doi.org/10.5555/3001460.3001507>
 132. Ikonomatakis N, Plataniotis K, Zervakis M, Venetsanopoulos A (1997) Region Growing and Region Merging Image Segmentation. In Proceedings of 13th International Conference on Digital Signal Processing, vol 1. IEEE, pp. 299–302. <https://doi.org/10.1109/ICDSP.1997.628077>
 133. Kang C, Wang F, Zong M, Cheng Y, Lu T (2020) Research on improved region growing point cloud algorithm. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLII-3/W10:153–157*. <https://doi.org/10.5194/isprs-archives-XLII-3-W10-153-2020>
 134. He L, Ren X, Gao Q, Zhao X, Yao B, Chao Y (2017) The connected-component labeling problem: a review of state-of-the-art algorithms. *Pattern Recognit* 70:25–43. <https://doi.org/10.1016/j.patcog.2017.04.018>
 135. Trevor A, Gedikli S, Rusu RB, Christensen HI (2013) Efficient organized point cloud segmentation with connected components. *Semant Perception Mapp Exploration (SPME) I*
 136. Liu W, Sun J, Li W, Hu T, Wang P (2019) Deep learning on point clouds and its application: a survey. *Sensors* 19(19):4188. <https://doi.org/10.3390/s19194188>
 137. Bello SA, Yu S, Wang C, Adam JM, Li J (2020) Review: deep learning on 3D point clouds. *Remote Sens* 12(11):1729. <https://doi.org/10.3390/rs12111729>
 138. Guo Y, Wang H, Hu Q, Liu H, Liu L, Bennamoun M (2020) Deep learning for 3D point clouds: a survey. *IEEE T Pattern Anal* 43(12):4338–4364. <https://doi.org/10.1109/TPAMI.2020.3005434>
 139. Qi CR, Su H, Mo K, Guibas LJ (2017) PointNet: deep learning on point sets for 3D classification and segmentation. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 77–85. <https://doi.org/10.1109/CVPR.2017.16>
 140. Qi CR, Yi L, Su H, Guibas LJ (2017) PointNet++: deep hierarchical feature learning on point sets in a metric space. In Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, pp 5105–5114. <https://doi.org/10.5555/3295222.3295263>
 141. Thomas H, Qi CR, Deschaud J-E, Marcotegui B, Goulette F, Guibas LJ (2019) Kpconv: flexible and deformable convolution for point clouds. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp 6410–6419. <https://doi.org/10.1109/ICCV.2019.00651>
 142. Hu Q, Yang B, Xie L, Rosa S, Guo Y, Wang Z, Trigoni N, Markham A (2020) RandLA-Net: efficient semantic segmentation of large-scale point clouds. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 11105–11114. <https://doi.org/10.1109/CVPR42600.2020.01112>
 143. Behley J, Garbade M, Milioto A, Quenzel J, Behnke S, Stachniss C, Gall J (2019) SemanticKITTI: a dataset for semantic scene understanding of LiDAR sequences. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp 9296–9306. <https://doi.org/10.1109/ICCV.2019.00939>
 144. Boulch A, Puy G, Marlet R (2020) Fkaconv: feature-kernel alignment for point cloud convolution. In *Computer Vision – ACCV 2020: 15th Asian Conference on Computer Vision, Kyoto, Japan, November 30 – December 4, 2020, Revised Selected Papers, Part I*, Springer, Berlin, Heidelberg, 381–399. https://doi.org/10.1007/978-3-030-69525-5_23
 145. Xiu H, Liu X, Kim T, Kim K-S (2025) Advancing ALS applications with large-scale Pre-training: framework, dataset, and downstream assessment. *Remote Sens* 17(11):1859. <https://doi.org/10.3390/rs17111859>
 146. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of Deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North

- American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
147. Xie Z, Zhang Z, Cao Y, Lin Y, Bao J, Yao Z, Dai Q, Hu H (2022) SimMIM: a simple framework for Masked image modeling. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 9643–9653. <https://doi.org/10.1109/CVPR52688.2022.00943>
 148. Hess G, Jaxing J, Svensson E, Hagerman D, Petersson C, Svensson L (2023) Masked autoencoder for self-supervised Pre-training on LiDAR point clouds. In 2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), pp 350–359. <https://doi.org/10.1109/WACVW58289.2023.00039>
 149. Lin Z, Wang Y, Qi S, Dong N, Yang M-H (2024) BEV-MAE: Bird's eye view Masked autoencoders for point cloud Pre-training in autonomous driving scenarios. In Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence. <https://doi.org/10.1609/aaai.v38i4.28141>
 150. Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: the KITTI dataset. *Int J Robot Res* 32(11):1231–1237. <https://doi.org/10.1177/0278364913491297>
 151. Borgmann B, Hebel M, Arens M, Stilla U (2017) Detection of persons in MLS point clouds. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLII-2/W7:203–210*. <https://doi.org/10.5194/isprs-archives-XLII-2-W7-203-2017>
 152. Sevgen E, Abdikan S (2023) Point-wise classification of high-density UAV-LiDAR data using gradient boosting machines. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLVIII-M-1-2023:587–593*. <https://doi.org/10.5194/isprs-archives-XLVIII-M-1-2023-587-2023>
 153. Yao Z, Nagel C, Kunde F, Hudra G, Willkomm P, Donaubaer A, Adolphi T, Kolbe TH (2018) 3DCityDB - A 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data Softw Stand* 3(1):1–26. <https://doi.org/10.1186/s40965-018-0046-7>
 154. Williams F, Gojic Z, Khamis S, Zorin D, Bruna J, Fidler S, Litany O (2022) Neural fields as learnable kernels for 3D reconstruction. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 18479–18489. <https://doi.org/10.1109/CVPR52688.2022.01795>
 155. Remondino F, El-Hakim S, Girardi S, Rizzi A, Benedetti S, Gonzo L (2009) 3D virtual reconstruction and visualization of complex architectures - the 3D-ARCH Project. *Int Arch Photogramm Remote Sens Spatial Inf Sci* 38(5/W1)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.