PBF-FR: Partitioning Beyond Footprints for Façade Recognition in Urban Point Clouds

Chiara Romanengo [1], Daniela Cabiddu [1], Michela Mortara

Istituto di Matematica Applicata e Tecnologie Informatiche "E. Magenes", Consiglio Nazionale delle Ricerche

Abstract

The identification and recognition of urban features are essential for creating accurate and comprehensive digital representations of cities. In particular, the automatic characterization of façade elements plays a key role in enabling semantic enrichment and 3D reconstruction. It also supports urban analysis and underpins various applications, including planning, simulation, and visualization. This work presents a pipeline for the automatic recognition of façades within complex urban scenes represented as point clouds. The method employs an enhanced partitioning strategy that extends beyond strict building footprints by incorporating surrounding buffer zones, allowing for a more complete capture of façade geometry, particularly in dense urban contexts. This is combined with a primitive recognition stage based on the Hough transform, enabling the detection of both planar and curved façade structures. The proposed partitioning overcomes the limitations of traditional footprint-based segmentation, which often disregards contextual geometry and leads to misclassifications at building boundaries. Integrated with the primitive recognition step, the resulting pipeline is robust to noise and incomplete data, and supports geometry-aware façade recognition, contributing to scalable analysis of large-scale urban environments.

Keywords: Geometric Primitive Fitting, Hough Transform, Large-Scale Urban Analysis, Façades Recognition, Point Cloud Partitioning

1. Introduction

Recognition of urban elements plays a key role in the development of detailed and reliable digital representations of cities, particularly within the framework of urban digital twins. These dynamic, high-fidelity virtual models of cities integrate geometric, semantic, and temporal information to support decision-making, analysis, and simulation across a range of urban applications. In this context, the automatic identification and characterization of façade elements is essential for enabling semantic enrichment and 3D reconstruction, supporting urban analysis, and informing downstream tasks such as simulation, planning, and visualization.

This paper contributes to this line of research by focusing on façade-level recognition within complex urban scenes. Most previous works address façade recognition at the scale of individual buildings. They typically begin by partitioning a large-scale input point cloud into separate building instances. This segmentation is often carried out using available footprint information, which serves as a spatial reference to isolate each building from its surroundings before applying further analysis. Specifically, points are assigned to buildings according to the position of their projection onto the plane where building footprints lie. Nevertheless, since footprint polygons are a rough representation of the shape and spatial extent of buildings, this strategy often overlooks important contextual information, particularly in dense urban areas like historical centres, where buildings are adjacent and may share walls. Besides, footprints in open data portals are not validated and might present various issues (imprecise position, wrong connections, self-intersections, etc.). As a result, points near boundaries or outside the strict footprint area may be misclassified or discarded, limiting the completeness and accuracy of the recognition process. Simply applying a uniform offset to the footprint is not a viable solution, as it risks clipping parts of adjacent buildings. Indeed, determining an appropriate offset value is nontrivial: it would require analyzing the surroundings of each individual building to tailor the offset accordingly.

Our contribution aims to overcome these limitations by introducing a streamlined pipeline that couples an improved spatial partitioning strategy with a primitive fitting approach based on the Hough Transform (HT). Together, these components address the challenges of isolating buildings and recognizing façade structures in dense and heterogeneous urban contexts.

The first stage of the pipeline automatically reorganizes the input data into spatially contiguous and non-overlapping Regions of Influence (RoIs). Each region includes one or more buildings, possibly adjacent to each other, as well as a surrounding buffer zone derived from their footprints. Note that the buffer zone has no fixed size or thickness, but conversely, it depends on the arrangements of neighbouring buildings, similarly to a Voronoi tessellation. This enhanced partitioning strategy captures spatial continuity and structural relationships that are often lost in traditional footprint-based segmentation. As a result, the method reduces misclassifications at boundaries and ensures that contextual geometry is preserved for further analysis.

In the second stage, we apply a primitive fitting approach to detect façade elements within each region. Using the Hough

*Corresponding author

Email address: chiara.romanengo@ge.imati.cnr.it (Chiara Romanengo [1])

[1]C. Romanengo and D. Cabiddu contributed equally to this paper.

Transform, the pipeline identifies both planar and curved primitives directly from the point cloud. This technique is robust to noise and missing data, common in LiDAR acquisitions, and is capable of detecting complex architectural features such as curved façades and church elements.

By integrating context-aware partitioning with robust primitive detection, our pipeline supports a more complete and accurate interpretation of the built environment, with a specific focus on façade recognition in complex urban scenes, contributing directly to the advancement of urban digital twins. The proposed contribution is threefold:

- We introduce an adaptive partitioning to segment the urban point cloud into chunks containing single buildings or a group of few adjacent buildings. More than setting a fixed size buffer zone, this reduces the risk of cutting away building points or, conversely, including points from adjacent buildings, which can happen where buildings are very close. Besides, the partitioning allows downstream parallelization;
- We perform automatic recognition of façades, our method being able to identify both planar and curved walls;
- Our approach retrieves the parametric form of a planar or cylindrical surface fitting a façade; therefore, it is straightforward to re-sample points, e.g., for simplifying huge models, for homogenising the local resolution, or for refining poorly sampled façades, as it is often the case with aerial surveys.

In the following, we first introduce the basic notions and terminology used throughout the paper (Section 2), and then review relevant previous work (Section 3). Our proposed approach is described in Section 4, followed by experimental results in Section 5. Finally, Section 6 concludes the paper and outlines directions for future work.

2. Basic notions and terminology

In this section, we summarize some notions and terminology from the remote sensing and civil engineering domains that are used in the paper.

Urban point clouds. Point clouds representing an urban area can be obtained by different acquisition techniques. First, *photogrammetry*, introduced more than a century ago by now, is widely used as a reality-capture technique, and it is likely the most popular. It builds on the relation between reality and its projection on the image space, and uses several overlapping images captured from multiple angles to generate 3D point clouds. Photogrammetry surveys can be conducted with airborne cameras, or even by satellites, and usually provide extensive mapping of urban areas.

Besides photogrammetry, *Light Detection and Ranging* (LiDAR) technology, a more recent acquisition technique, relies on laser scanning and measures the time it takes for light to travel to a surface and return to the sensor. LiDAR data can be obtained from aerial systems like drones or airplanes ALS – Aerial Laser Scanning) or terrestrial acquisitions, including mobile platforms (MLS – Mobile Laser Scanning) mounted on cars or portable devices used while walking. Terrestrial static setups (TLS – Terrestrial Laser Scanning) are also employed for detailed stationary scans. LiDAR data are provided in the standard LAS format [1] (or LAZ when compressed) and may include, in addition to the spatial coordinates of points, some attributes, such as a preliminary classification into categories like vegetation, buildings, and roads.

The deployed technology and survey setting influence the quality and characteristics of the acquired point cloud. For instance, ALS is efficient to acquire wide areas; it captures well top surfaces like roofs, while ground points and vertical surfaces might remain hidden depending on the flight direction. Conversely, terrestrial acquisition largely misses roofs; TLS achieves the highest resolution and accuracy, but the size of produced data and the time needed to move the scanner to many fixed locations limit the intervention area.

Level of detail. CityGML [2] is the international standard of the *Open Geospatial Consortium* (OGC) that defines the objects, attributes, and relationships required to set up a 3D city model application. CityGML organizes the description of 3D urban models according to the 3D geometry, topology (e.g., adjacency and hierarchies), semantics, and appearance properties. CityGML also includes the concept of Level of Detail (LOD), varying from LOD0 (less detailed level) to LOD4 (most detailed level) [3]. The coarsest level (LOD0) describes urban features as flat polygons embedded in 3D space. As an example, a building is depicted by its footprint polygon, with height equal to the ground or to the maximum height of the building itself. Differently, LOD1 identifies urban features as volumetric objects; for instance, a building is represented as a blocks model without any roof structures. LOD2 adds further details to buildings, like distinctive roof structures and larger installations like balconies and stairs. LOD3 includes also architectural and detailed wall and roof structures, doors, windows, and bays. Finally, LOD4 adds also interior details like rooms, stairs, and furniture. An illustration of the five LODs for a building is provided in Figure 1.

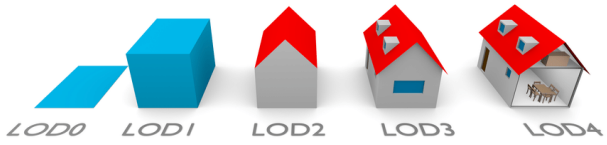


Figure 1: An illustration of LODs defined by CityGML. Courtesy of [4].

3. Previous work

Creating accurate and comprehensive digital representations of urban environments relies on the robust recognition and extraction of key features, with building façades playing a central role in 3D reconstruction, semantic enrichment, and various urban analysis applications. Central to this process is the partitioning of large-scale point clouds into meaningful subregions and the recognition of façade structures, both of which have been extensively addressed in the literature.

3.1. Point Cloud Partitioning

Partitioning is a key pre-processing step in managing large-scale point clouds, especially in urban environments, where isolating individual structures or functional areas is critical for

downstream tasks such as reconstruction or semantic analysis. Traditional geometric partitioning approaches rely primarily on local spatial continuity and regularity in shape. Region-growing methods, for example, incrementally build segments by merging neighboring points with similar geometric properties, such as normals or curvature (see [5]). Clustering techniques based on Euclidean distance or density, such as DBSCAN, are frequently employed due to their simplicity and robustness to noise, although they often lead to over-segmentation in dense or structurally complex areas [6].

Another common line of work explores the use of projection and slicing strategies. By projecting the 3D cloud onto 2D planes or slicing it along principal directions, these methods exploit the mostly planar nature of man-made environments to simplify segmentation. This approach is particularly effective in urban settings where buildings and streets follow well-defined orientations. Planar segmentation techniques, such as those based on RANSAC (RANdom SAmple Consensus), further capitalize on the prevalence of flat surfaces by extracting dominant planes, although they can struggle in cluttered or irregular scenes [7].

More scalable solutions often rely on hierarchical or graph-based representations. Structures like octrees allow for efficient spatial indexing and neighborhood queries, supporting large-scale processing workflows [8]. Supervoxel segmentation, which groups points based on local connectivity and feature similarity, is also widely used to provide an over-segmented representation of the scene [9]. Graph partitioning techniques built on such representations optimize cuts based on continuity or spectral properties, allowing for flexible adaptation to varying scene complexity.

Su et al. [10] introduce a two-stage approach for building point cloud extraction that relies solely on geometric features, achieving high accuracy without requiring external data or machine learning. Their results highlight the effectiveness of purely geometric cues in identifying façade structures.

Recently, hybrid methods that combine geometric heuristics with learning-based models have gained traction. These methods typically use geometric partitioning to generate initial over-segmentations, which are subsequently refined using semantic cues or learned feature embeddings. Techniques like PointNet [11] and Superpoint Graphs [12] have demonstrated the benefits of integrating geometric structure with contextual reasoning, though they often require substantial labeled datasets and computational resources.

3.2. Primitive detection on generic point clouds

In various application contexts, a wide range of methods have been developed to decompose digitized point clouds or meshes into regions approximated by geometric primitives. Following the taxonomy proposed in [13], these approaches can be categorized into four groups: stochastic methods, parameter space methods, clustering techniques, and learning-based approaches. A representative method in the first category is RANSAC [14]. The second group includes Hough-like voting schemes and parameter space clustering methods, such as the one described in [15]. The third category encompasses a variety of clustering techniques, including primitive-driven region growing (e.g., [16]) and primitive-oblivious segmentation (e.g., [17]). Finally, supervised fitting methods have also been proposed, even for multi-class primitive detection [18, 19]. The survey in [13] offers a comprehensive historical overview of simple primitive detection techniques, which is beyond the scope of this paper.

A recent approach for recognizing geometric primitives is described in [20], where a triangle mesh is subdivided into maximal surface portions using a curvature-based partitioning method. Subsequently, [21] introduces a region-growing system for segmenting large point clouds into planar regions. Other methods, which are specifically designed to detect certain types of primitives, include: [22], which handles cylinders and planes; [23], which focuses on quadric surfaces; [24], which fits surfaces of revolution; and [25], which extracts cylindrical shapes from non-oriented point clouds.

Regarding deep learning-based methods, ParSeNet [26] and HPNet [27] have shown promising results in the problems of segmentation and recognition of primitives, able to handle both simple primitives and open and closed spline surfaces. Another learning-based approach is PriFit [28], which is able to decompose a point cloud of different 3D shape categories into a collection of geometric primitives, such cuboids and ellipsoids, or similarly deformed planes. However, the natural flexibility of supervised learning approaches in identifying geometric primitives comes with a considerable cost: the requirement for massive labelled training sets, whose collection challenges restrict their current use.

3.3. Primitive detection on urban LiDAR point clouds

Focusing on the urban context, the literature presents a range of methods for the identification of high level urban features, based on point classification in categories such as building, vegetation, road and other general types, similarly to the LiDAR classification, e.g., [29, 30]. Other approaches tackle a mixed selection of features, such as façades, ground, cars, motorcycles, traffic signs, pedestrians, vegetation in [31]. Some methods tackle specific features, e.g., roofs, with the primary aim of reconstructing buildings rather than an accurate urban element recognition [32, 33]. Similarly, the techniques described in [34, 35, 36] also deal with the detection of roof planes and the optimization of roof edges in order to reconstruct a building. Finally, some supervised learning approaches (e.g., [37, 38]), rely on additional information that can be embedded within the point cloud, such as intensity or RGB color. However, the availability of this data largely depends on the acquisition technology and may not always be present.

For a more complete overview of existing methodologies for geometric primitives in LiDAR point clouds, we refer to [39].

3.4. Façade Detection

The extraction and recognition of building façades is a key challenge in the field of 3D urban modeling, and it has been explored from multiple perspectives in the literature. Depending on the goal, different methods have been proposed. For example, some works focus specifically on the reconstruction or modeling of façades [40, 41, 42], while others aim at recovering entire buildings [43, 44, 45, 46]. In this context, [40] introduces a review of façade reconstruction methods with the aim of highlighting data and key technologies used in the literature to enrich building façades and focusing on approaches for façade parsing and building-opening detection.

Different methods exploit the identification of façades for extracting some features, such as façade openings or balconies

[47, 48, 49], and for creating LOD representations [50, 51]. For instance, [47] introduces an algorithm to detect boundaries of building façades and their openings. Similarly, [42] identifies repeating structures within a single façade scan, and use patterns of structures to model façades and their features neatly. However, the façade recognition is not involved in the process. Wang et al. [48] proposes a façade feature extraction that combines a 3D point cloud with optical images considering structural information. The extraction of façade features often helps the creation of LOD representation. For instance, [51] introduces a method for generating LOD3 building models. The pipeline consists of two main components: the first one performs deep-learning-based image segmentation to identify openings, opening corners, or facades, while the second step generates LOD2 models from point clouds, which are then upgraded to LOD3 using the information about openings. Indeed, the majority of learning-based methods focuses on the extraction of façade features, exploiting image data in addition to point clouds. The review in [52] explores trends and relationships in AI-driven facade research, interested in architectural, environmental, and structural aspect of façade design, maintenance, and operation. The mentioned papers focus on the identification of façade details which are outside the scope of our work, since we are interested in the global façade structure.

Finally, semantic segmentation of 3D point clouds plays a crucial role in urban reconstruction and façade recognition. Several methods have been proposed to classify and extract building elements based on geometric and spatial information. In particular, Romanengo et al. [53] propose a semantic segmentation approach for large-scale point clouds based on primitive recognition, under the assumption that each building is entirely defined by its footprint. Although they slightly offset footprint polygons to improve alignment, their method does not explicitly account for spatial context or the continuity between adjacent buildings in dense urban settings.

In contrast, our method integrates spatial reasoning directly into the segmentation process, using an enhanced partitioning strategy that incorporates buffer zones around building footprints. This allows us to better preserve contextual relationships and minimize errors at building boundaries, ultimately enabling more reliable façade recognition in complex urban scenes.

Differently, the approach described in [54] use convolutional neural networks applied to aerial LiDAR point clouds, extracting predictors from neighbouring geometric features at each point with the strategy proposed in [55]. The goal of this approach is to classify the points into two semantic classes, roofs and façades.

Table 1 summarizes the main characteristics of the methods introduced in this section, focusing on the type of input, the recognition target (whole façade versus detail architectural elements), the type of façade detected (planar or curved) and the ability to resample the façade itself.

4. Our approach

Our method takes as input a 3D point cloud representing the urban area of interest, along with a 2D polygonal representation of building footprints, typically provided in standard GIS formats such as ESRI Shapefiles, GeoJSON, or GeoPackage. Optionally, the boundary of the study area may also be specified.

The first stage of the pipeline involves reorganizing the input data into spatially contiguous and non-overlapping Regions of Influence (*RoIs*), which are automatically computed (see Section 4.1). Each RoI refers to (and includes) one or more buildings, possibly adjacent to one another, as well as a surrounding buffer zone derived from their footprints. The size of the buffer zone is not fixed a priori, but constructed to include points nearer to that building(s) than to all the others. This improved partitioning strategy captures the spatial continuity and relationships between structures, ensuring full data coverage and preserving contextual information often lost in footprint-based segmentation.

In the second stage, we apply a primitive fitting strategy to identify façade elements within each region (see Section 4.2). Using the Hough Transform, we detect both linear and curvilinear primitives directly from the point cloud. This technique is particularly robust to noise and missing data, making it well-suited to the characteristics of real-world LiDAR scans. It enables the recognition of a wide variety of façade geometries, including complex architectural features such as curved walls or church elements.

By integrating context-aware partitioning with robust primitive detection, our pipeline reduces misclassifications at building boundaries and, thus, supports a more complete and accurate interpretation of the urban context.

4.1. Point cloud partitioning

Our partitioning strategy begins with a set of input building footprints and, optionally, the external boundary of the area of interest. If the external boundary is not provided, it can be set either manually or automatically as the convex hull or as the axis-aligned bounding box of all footprints. Based on this information, we automatically construct a subdivision of the domain into building-wise RoIs, where each RoI is designed to include one or a few adjacent buildings along with a surrounding buffer zone.

The pipeline to build the subdivision into RoIs is shown in Figure 2. We first compute a constrained Delaunay triangulation of the area, where the edges of the building footprints are enforced as constraints to preserve the geometry of each structure and their interiors are treated as holes, preventing triangles from being generated within the footprint areas (see Figure 2(b)). From this triangulation, we construct the dual graph joining the centroid of each triangle to the midpoints of its interior edges, while ignoring mesh boundary edges (see Figure 2(c)).

To convert this dual graph into a collection of polygonal RoIs suitable for guiding the partitioning of the input point cloud, we generate a second constrained triangulation, where the edges of the dual graph are used as constraints (see Figure 2(d)). A region growing algorithm based on a breadth-first search (BFS) is then applied to determine maximal connected regions (see Figure 2(e)). Each RoI corresponds to a polygonal area that serves as a spatial unit for assigning points. Points within a RoI are nearer to the reference buildings than to any other: intuitively, the dual graph approximate the Medial Axis Transform and the RoIs mimic the Voronoi cells for buildings. The RoI collection is a gap-free and spatially consistent subdivision of the domain, preserving both building footprints and their surrounding context.

We use the RoIs to guide the point cloud partitioning, which relies on the point-in-polygon test. For a detailed discussion of

Method	Input	Global Structure	Architectural Features	Planar	Curved	Resampling
Wang et al. 2016 [42]	LiDAR PC (TLS) of a single façade	yes	yes	yes	yes *	yes
Truong et al. 2014 [47]	LiDAR PC (TLS)	yes	no	yes	no	no
Wang et al. 2018 [48]	2D optical images + LiDAR PC	yes	yes	yes	no	no
Hu et al. 2024 [49]	LiDAR PC	yes	yes	yes	yes	no
Wang et al. 2023 [50]	Photogrammetric PC	yes	no	yes	no	no
Pantoja-Rosero et al. 2022 [51]	Multiple-view images	yes	yes	yes	no	no
Romanengo et al. 2024 [53]	LiDAR PC (ALS), building footprints	yes	no	yes	yes	no
Pirotti et al. 2019 [54]	LiDAR PC (ALS)	yes	no	yes	no	no
Ours	*LiDAR PC (ALS), building footprints*	*yes*	*no*	*yes*	*yes*	*yes*

Table 1: Summary of the main characteristics of façade detection methods introduced in Section 3.4. For each method, the table reports the type of input data used (PC stands for point cloud), whether the method captures the overall structure and/or architectural features, the type of façades it can detect (planar and/or curved), and whether it supports façade resampling. An asterisk (*) indicates that the method supports only curved façades that can be represented as developable surfaces.

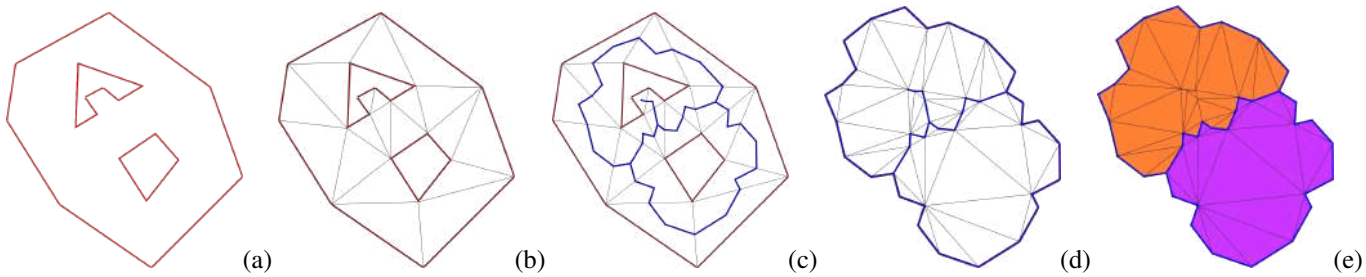


Figure 2: Pipeline for automatically generating the subdivision into RoIs to guide building-wise point cloud partitioning. (a) Input building footprints and overall domain boundary. (b) Constrained Delaunay triangulation of the area, where building footprints are treated as holes. (c) Dual graph (in blue) constructed by connecting triangle centroids and edge midpoints. (d) Constrained triangulation of the dual graph. (e) Final RoIs obtained by region growing via breadth-first search.

the point-in-polygon problem and various approaches, we refer to the literature [56]. The algorithm is based on the Jordan Curve Theorem [57], which states that a point p lies inside a polygon if a half-line extending from p in any arbitrary direction crosses the polygon's boundary an odd number of times. Our implementation extends the method introduced by W. Randolph Franklin [58], enhancing it to support both simple polygons and polygons with multiple boundaries, including holes.

We apply this approach, potentially in parallel, to the entire set of RoIs to assign each point to its corresponding spatial region. Specifically, for each point in the cloud, we perform a point-in-polygon test on its projection onto the 2D plane of the RoI subdivision, e.g. $z = 0$. If the point falls within one of the generated RoIs, it is assigned to it.

The final output of this step is a set of point chunks, each corresponding to a RoI and containing all the points assigned to it. These chunks serve as spatial units for the façade recognition approach based on fitting primitives and Hough transform.

4.2. Façade recognition

After the partitioning, each chunk of points is translated to the origin to avoid numerical issues due to the geographical coordinates. Then, every chunk is divided into subsets of points $\mathcal{P}$ representing planes or cylinders. We experimented two approaches: RANSAC [14] and fitting primitives by region growing [5], both implemented in CGAL [59]. While RANSAC recognizes both plane and cylinder instances, the region growing approach provides different applications for identifying one primitive or another. However, it provides more precise results than RANSAC, and in turn it is more time consuming. In this implementation, we aim at maximum precision and use the latter [5]. Since the algorithm requires the normal vectors associated to the point cloud, we apply beforehand a specific algorithm in [60] that computes normals from the LAS properties typically provided by LiDAR sensors.

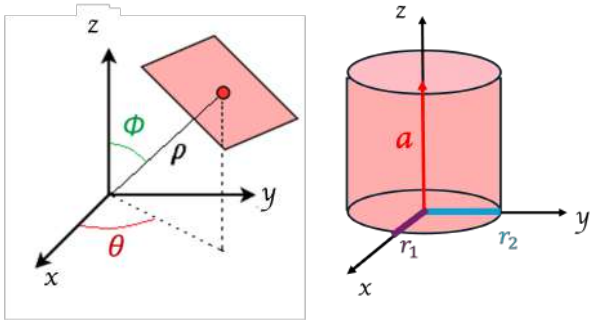
Then, each segment $\mathcal{P}$ is processed by a recognition approach based on the Hough transform (HT), since it is a technique well-known for its robustness to noise and missing parts [61], from which point clouds acquired by laser scanners are often affected. The use of the HT was originally limited to the recognition of straight lines in images [62]; then it has been extended to a wide range of algebraic primitives in [63] and, recently, generalized to surface primitives in [64]. In our context, we are interested in finding the surface (plane or cylinder) that best fits a given subset of points $\mathcal{P}$. More in detail, once we fix a family of surfaces $\mathcal{F} = \{\mathcal{S}_{\mathbf{a}}\}$ that depends on a set of parameters $\mathbf{a} \in \mathbb{R}^n$, the problem is to find a set of parameters $\bar{\mathbf{a}} = (a_1, \ldots, a_n)$ such that the surface $\mathcal{S}_{\bar{\mathbf{a}}}$ best fits $\mathcal{P}$. This result is obtained by the HT exploiting the duality between $\mathcal{P}$ and the parameters space $\mathbb{R}^n$, since a point $\mathbf{p}$ in $\mathcal{P}$ corresponds to a locus $\Gamma_{\mathbf{p}}(\mathcal{F})$ in $\mathbb{R}^n$. The duality concept translates the recognition problem into detecting which values of the parameters of the family $\mathcal{F}$ correspond to the surface that best fits the set of points $\mathcal{P}$. Varying $\mathbf{p}$ on $\mathcal{P}$, a set of loci $\{\Gamma_{\mathbf{p}}\}$ is generated and the intersection of $\{\Gamma_{\mathbf{p}}\}$ contains the parameters $\bar{\mathbf{a}}$ that generate the best fitting surface $\mathcal{S}_{\bar{\mathbf{a}}}$. In practice, this theoretical aspect is translated on the so-called *accumulator matrix*, that consists in a voting system whereby each point $\mathbf{p}$ in $\mathcal{P}$ votes for a n-uple $\mathbf{a} = (a_1, \ldots, a_n)$ and the most voted n-uple is selected as the most representative surface for $\mathcal{P}$. The robustness of the HT is inherited by the voting procedure.

In our context, we fix two family of surfaces, that is planes and cylinders, since normally a façade of a building is represented by a plane or part of a cylinder in the case of some par-

ticular structures, such as churches. Regarding the family of planes $\mathcal{F} = \{\mathcal{S}_{\mathbf{a}}\}$, we use the mathematical expression in the Hesse normal representation:

$$x\cos\theta\sin\phi + y\sin\theta\sin\phi + z\cos\phi - \rho = 0,$$

in which the parameters $\mathbf{a} = (\theta, \phi, \rho)$ are the polar coordinates of the normal vector $\mathbf{n} = (\cos\theta\sin\phi,\ \sin\theta\sin\phi,\ \cos\phi)$ to the plane $\theta \in [0, 2\pi)$, $\phi \in [0, \pi)$ and the distance $\rho \geq 0$ from the plane to the origin of the coordinate system.



(a) (b)

Figure 3: Visual representation of parameters for plane in (a) and cylinder in (b).

For the family of cylinders $\mathcal{F} = \{\mathcal{S}_{\mathbf{a}}\}$, we use the parametric representation in standard form

$$\begin{cases} x(u,v) = r_1\cos u \\ y(u,v) = r_2\sin u \\ z(u,v) = v \end{cases} \qquad u \in [0, 2\pi),\ v \in \mathbb{R}$$

where the parameters $\mathbf{a} = (r_1, r_2)$ correspond, respectively, to the horizontal and vertical semi-axes of the ellipse at the base of the cylinder. In this representation, we assume that the axis of the cylinder $\mathbf{a}$ coincides with the $z-$axis, since a cylinder in general position in the space depends on at least 9 parameters and the number of parameters for the HT directly influences the size of the accumulator matrix. This reduction in parameters makes it possible to apply HT to the case of cylinders that would otherwise not be affordable in practice due to the exploding dimension of the accumulator matrix. The assumption requires a pre-processing of $\mathcal{P}$ to put it in the standard form, by rotating and/or translating it. To do this, we exploit the strategy in [64] that computes initial estimates for the parameters and localizes the search for the optimal solution. Once the recognition is performed, transformations are applied backward to obtain the coordinates of the cylinder axis $\mathbf{a}$ and its position in the space. Figure 3 provides a visual representation of parameters for both plane (a) and cylinder (b).

The result of the HT is, for planes, the polar coordinates of the normal vector $\bar{\mathbf{a}} = (\bar{\theta}, \bar{\phi}, \bar{\rho})$, and for cylinders, the horizontal and vertical semi-axes of the ellipse at the base of the cylinder $\bar{\mathbf{a}} = (\bar{r}_1, \bar{r}_2)$ with the coordinates of is axis $\bar{\mathbf{a}}$. At this point, we exploit these parameters to classify $\mathcal{P}$ as a façade:

- if $\mathcal{P}$ is a plane, we first update $\phi = \pi - \bar{\phi}$ if $\bar{\phi} \in [\frac{\pi}{2}, \pi]$ and then identify $\mathcal{P}$ as façade if $\frac{\pi}{2} - \varepsilon < \bar{\phi} \leq \frac{\pi}{2}$, since $\bar{\phi}$ corresponds to the inclination of the recognized plane with respect to the horizontal plane;
- if $\mathcal{P}$ is a cylinder, we identify $\mathcal{P}$ as façade if $\mathbf{a}$ coincides with the $z-$axis.

In our experiments, we empirically fix $\varepsilon = \frac{\pi}{36}$ because of the large amount of noise typical of the input data type.

Since the HT provides the equation of recognized surface, we exploit it in two different ways. In case the chunk derived from a airborne acquisition, we use the parametric representation to complete the missing parts of each recognized façade, since this type of acquisition cannot completely capture some parts of buildings due to its nature. If the chunk derived from a terrestrial acquisition, we use the equation to resample the façade, since they are often represented by an excessive amount of points. More in details, we can obtain the vertices of the border polygon of each façade in order to produce a LOD1 representation of the urban area.

5. Experimental results

The proposed pipeline was implemented in C++ to guarantee both computational efficiency and flexibility. For geometric processing, we utilized the CGAL library [59] for fitting primitives, and the Cinolib [65], which offers a robust framework for managing polygonal meshes, for computing dual graphs, and performing operations like meshing and region adjacency queries. In parallel, the GDAL library [66] was used to manage geospatial data, particularly for reading and processing building footprints stored in vector formats (e.g., shapefiles). *The source code will be released upon acceptance.*

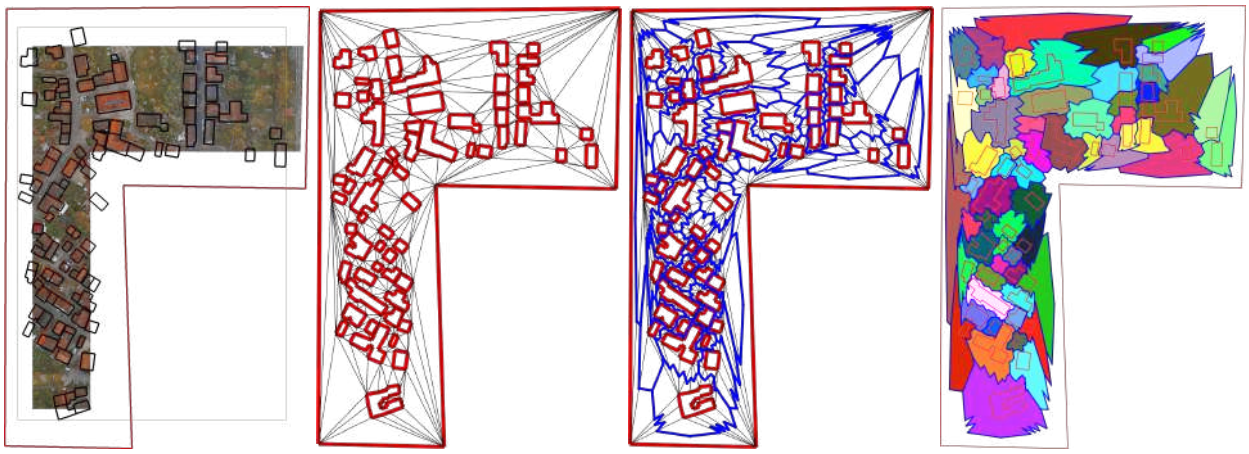
5.1. Datasets

To validate the effectiveness of our method, we first conducted a series of experiments on a subset of the H3D dataset [67]. We selected the training dataset, because it has been manually annotated. In particular, we refer to the "facade" label as ground truth for evaluating the accuracy of our method. The selected point cloud contains approximately 63 million points. The data acquisition has been performed via aerial autonomous unmanned vehicle (AUV).

Building footprints were extracted from OpenStreetMap (OSM) [68]. OSM provides a lightweight and accessible source of 2D reference geometry; however, OSM footprints are user-created and not validated, hence, they may be imprecise (see Figure 5) or contain topological inconsistencies. To define the spatial extent of our experiments, we manually delineate an external boundary of the study area covered by the LiDAR survey, ensuring a consistent and representative case study. Figure 4(a) shows the input point cloud, the building footprints at least partially covered by the LiDAR distribution, and the external domain boundary (here manually set). The dataset used for our experiments includes 100 building footprints.

In addition, we tested our methodology on aerial and terrestrial point clouds of 2 Italian cities with complex morphology, Genova and Catania, to show the advantage of the parametric representation for the recognized façades. In particular, we show the results on two specific buildings, one per city.

The Genova dataset is an aerial point cloud in LAS format, relative to the 2018 aerial acquisition downloaded from [69]. We experimented our method on the S. Marcellino church, a rationalist-style building. The volume of the church, with a cylindrical shape punctuated by large niches, is covered by a



(a) (b) (c) (d)

Figure 4: Spatial partitioning of the input point cloud. (a) Input LiDAR point cloud from the H3D dataset (November 2018 epoch). Building footprints from OpenStreetMap are overlaid in black, while the manually defined boundary of the overall study area is shown in red. (b) Constrained Delaunay triangulation of the domain, incorporating building footprints as constraints. (c) Corresponding dual graph, in blue. (d) The RoIs enabling and guiding the partitioning of the domain.

reinforced concrete dome, and the entrance is highlighted by a full-height monumental pronaos (see Figure 9(a)). These peculiar features make it challenging both from the acquisition and the recognition point of view. The building from Catania is a single high resolution point cloud acquired within the project *omitted for anonimity* through terrestrial long and mid range laser scanning systems, reaching a resolution smaller than 12 mm. Acquiring stations have been placed in 110 locations overall, at ground level and at higher floors, from the nearby alleys and on opposite buildings.

5.2. Evaluation, discussion and comparison

Evaluation on the H3D dataset. We first test our entire pipeline on the H3D dataset, starting with the point cloud partitioning. From the building footprints, we generate a constrained Delaunay triangulation that incorporates the 2D geometry of the scene, as shown in Figure 4(b). Then, we derive the corresponding dual graph (Figure 4(c)), which serves as the basis for the subsequent automatic definition of RoIs. As shown in Figure 4(d), the dual graph leads to a partition of 63 RoIs, as compared to the 100 building footprints. This indicates that some buildings are adjacent to each other and share common walls, at least from a footprint perspective.

To assess the accuracy of our façade recognition pipeline, we performed a quantitative evaluation using the annotated ground truth labels provided in the H3D dataset. Specifically, we considered a binary classification task, where each point in the input cloud is labeled as either façade or non-façade. This allows us to evaluate the results in terms of classical performance metrics: *precision*, *recall*, and *F1-score*. The classification results are analyzed by comparing our predicted labels with the ground truth, leading to the counts of *true positives* (TP), *false positives* (FP), *false negatives* (FN), and *true negatives* (TN).

Table 2 reports the detailed outcomes for the selected area.

Using these values, we computed the standard evaluation metrics:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

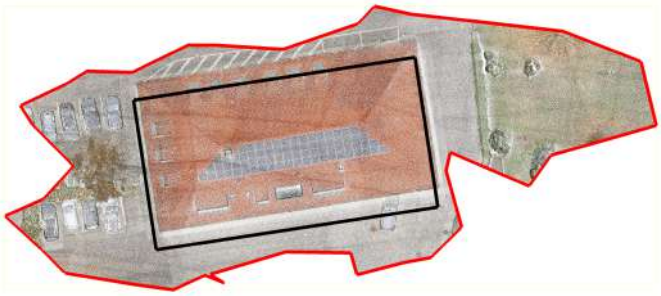


Figure 5: Top view of the point cloud contained within Chunk 54 of our partitioning. The footprint of the corresponding building is overlaid (in black) for comparison.

TP	TN	FP	FN
1,490,087	57,934,508	1,214,134	89,210

Table 2: Confusion matrix values computed over the selected evaluation area. TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives. These values form the basis for computing the precision, recall, and F1 Score discussed in the text (see Equations 1, 2 and 3).

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

From these values, we compute a precision of 0.551023 and a recall of 0.943513, resulting in an F1 Score of 0.69573.

Overall, the results confirm that our approach provides a robust and reliable tool for façade recognition, capable of handling complex scenes while maintaining a high level of completeness.

It is important to note that our definition of façade is more specific than the one adopted in the ground truth annotations.

In the ground truth, architectural elements such as windows, balconies, and doors are typically labeled as part of the façade. In contrast, our method deliberately excludes these elements, as we define the façade strictly as vertical structural planes (see Figure 6 and 7). This discrepancy results in a higher number of false negatives and a relatively low precision in our evaluation. However, these are not actual misclassifications, but rather a consequence of differing annotation criteria. Our more rigorous definition enables a clearer and more detailed semantic segmentation, which we believe offers added value in practical applications.

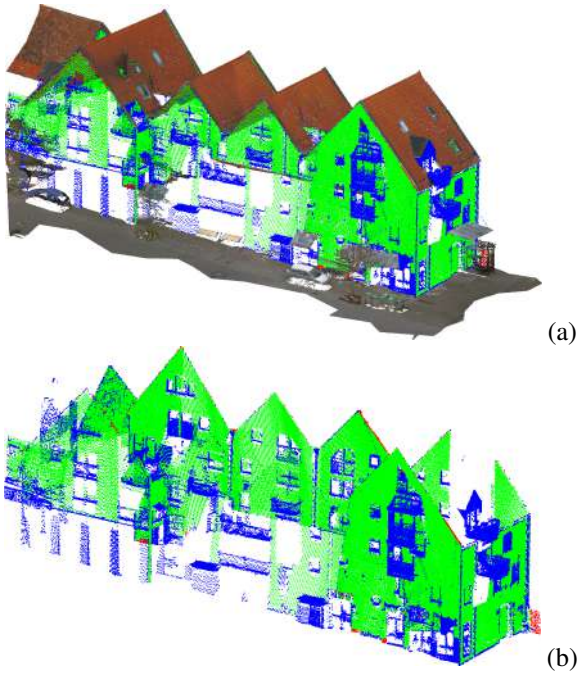


Figure 6: Example illustrating the discrepancy between our façade definition and the ground truth. (a) Point cloud resulting from our partitioning (Chunk 5 of the H3D dataset) with output of our façade recognition approach overlaid. (b) Output of our façade recognition approach. Green points represent true positives, while blue points indicate false positives. In the ground truth, elements such as windows, balconies, and doors are labeled as part of the façade, whereas our approach excludes them. This semantic difference may lead to a higher number of false positives, which do not reflect actual misclassifications but rather a more specific interpretation of what constitutes a façade.

Comparison with [53]. To compare the proposed methodology, we have select the most similar pipeline. Indeed, we have tested the approach in [53] on the H3D dataset and we have computed the same classification metrics of Equations 1, 2 and 3, obtaining a precision of 0.44835, a recall of 0.98733 and the resulting F1 score of 0.616677. We can observe that the proposed method demonstrates a more balanced trade-off between precision and recall compared to [53], as reflected by a higher F1-score (0.696853 with respect to 0.616677). Notably, the precision achieved by our approach (55.3%) represents a substantial improvement withe respect to [53] (44.8%), indicating a reduced rate of false positives and a greater reliability in positive predictions. While the recall of our method (94.2%) is slightly lower than that of the [53] (98.7%), it remains high, ensuring that the majority of true positive instances are correctly identified. The improved balance between precision and recall

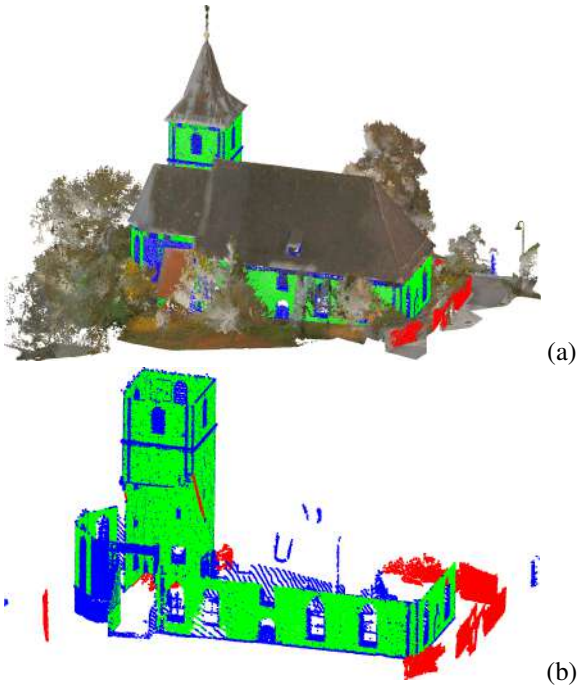


Figure 7: (a) Point cloud resulting from our partitioning (Chunk 48 of the H3D dataset) with output of our façade recognition approach overlaid. (b) Output of our façade recognition approach. Green points represent true positives, blue points indicate false positives, and red points correspond to false negatives. This example highlights two key aspects: (i) some regions, such as the perimeter wall surrounding the church (in red), are identified as façades by our method but not labeled as such in the ground truth, resulting in apparent false negatives due to differing definitions; (ii) despite the presence of noise in the chunk, the recognition remains robust and effective.

results in a more favorable F1-score, which serves as a robust indicator of the overall performance, particularly in scenarios where neither metric alone is sufficient to capture classification quality. We can conclude that our approach is better suited for real-world applications, where minimizing both false positives and false negatives is essential.

Façade recognition and parametric resampling of architectural elements. We experimented the approach on a challenging building in Genova to demonstrate the capability of recognizing curved walls and the robustness of our method over under-sampled data. In the first row of Figure 9, we can see the S. Marcellino church and the result of the spatial partitioning that extracts the corresponding chunk. The chunk is segmented into subsets of points representing planes and cylinders (Figure 9(IIa)); segments composing the walls are then identified and highlighted in grey in Figure 9(IIb); finally, the undersampled areas of each recognized façade are refined through the parametric representation of the best fitting cylinder, adding points where needed (Figure 9(IIc)). The identification of the parametric form of planar and cylindrical primitives fitting the façades indeed allows the straightforward re-sampling, including refining, and reconstruction of the missing parts of the recognised walls.

The resampling is of course possible for terrestrial data as well. This is mostly useful to control the local resolution, e.g., to coarsen planar patches with near lossless compression. An example is provided in Figure 10(a) where a point cloud rep-

Figure 8: Examples of façade recognition results on the H3D dataset. Each row corresponds to a different chunk. The number at the end of each line indicates the chunk identifier. The first column shows the point cloud resulting from the partitioning. The second column overlays the recognized façades on the original point cloud. The third column highlights only the recognized façades. Green points represent true positives, blue points indicate false positives, and red points correspond to false negatives.

resenting an historical building of Catania is shown. More in detail, in Figure 10(b) the recognized façades are overlaid to the original point cloud, while in Figure 10(c) the points corresponding to façades are downsampled by exploiting the parametric representation of planes.

5.3. Limitations

Our strategy relies on building footprints, which are often affected by noise and inconsistencies; an example are those extracted from OpenStreetMap, which are not officially verified. Our partitioning strategy is robust to slight misalignments between the footprints and the actual building positions. However, when footprints are significantly misaligned, our partitioning can fail (see Figure 11). In principle, failure happens when the displacement in the direction of an adjacent building is higher than half the distance between the two (α in Figure 12). Indeed, the dual graph for determining the RoIs acts as a medial structure, locally dividing the space between two buildings at midway. To cope with this issue, common when buildings are particularly dense, it is possible to group buildings closer than a certain distance, related to half the maximum expected displacement in the footprints and the minimum distance between footprints, but the proper threshold needs further investigation.

Since the partitioning strategy groups spatially contiguous regions based on footprint proximity and shared buffer zones (as in Figure 13), it does not explicitly preserve the separation between buildings that are directly adjacent. As a result, adjacent buildings may be considered as a single entity during the façade recognition process. Addressing this issue, possibly by incorporating additional cues such as height discontinuities or structural symmetry, remains an important direction for future work.

Additionally, we observe occasional misclassifications in the façade recognition step. Common false positives include vertical structures such as chimneys, tree trunks, or even the sides of cars, which may be incorrectly identified as planar surfaces due to their geometric regularity and vertical orientation (see false negatives highlighted in Figure 7). These errors could be mitigated by introducing geometric filters based on expected façade dimensions, connectivity, or planarity persistence. Further improvements might also be achieved by leveraging temporal consistency (in multi-epoch acquisitions) or contextual cues from surrounding structures.

6. Conclusions

This work presents a novel pipeline for façade recognition in complex urban scenes represented as point clouds. The method is designed to enhance robustness and completeness in challenging contexts, such as dense historical centers or areas with noisy and incomplete data. It combines two key components: (i) an improved spatial partitioning strategy that goes beyond

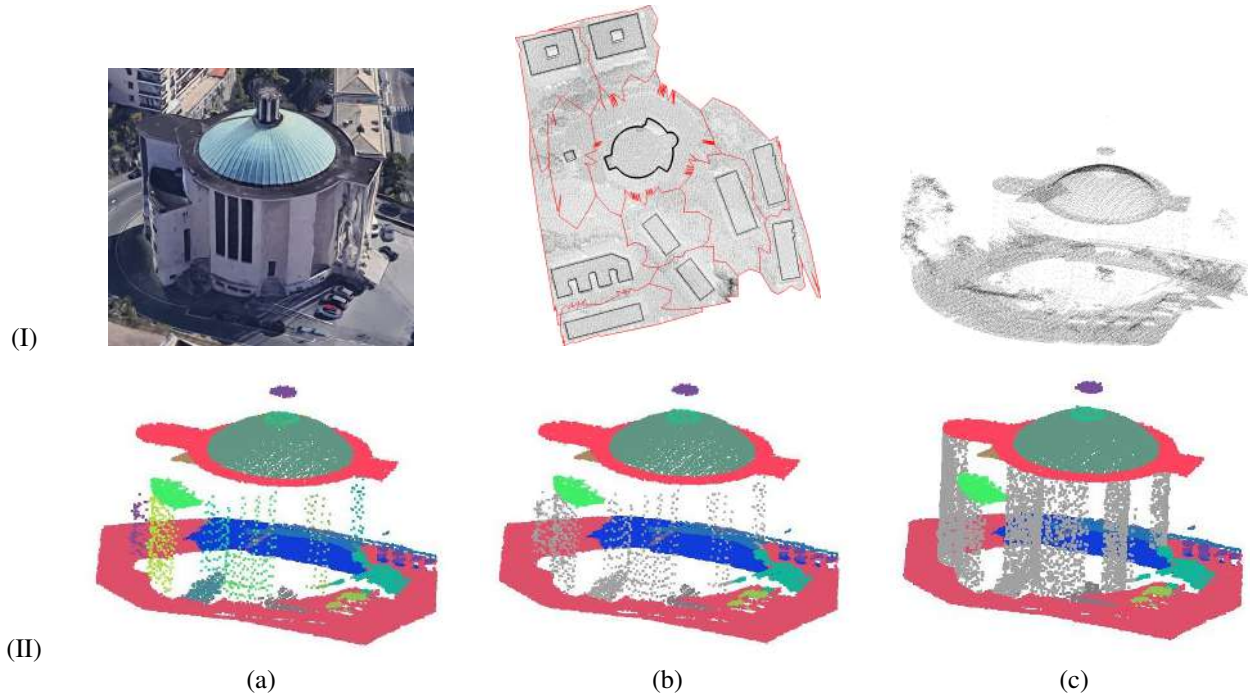


Figure 9: Airborne acquisition of the church of "San Marcellino" in Genova. First row (I) depicts the extraction of the chuck corresponding to the church: in (a) a real picture, in (b) the result of the spatial partitioning and in (c) the chuck. Second row (II) provides the steps of the facades recognition approach: in (a) the result of the segmentation, in (b) the recognition of cylinders corresponding to facades highlighted in grey and in (c) the missing parts of the recognized facades are reconstructed through the parametric representation of cylinders.

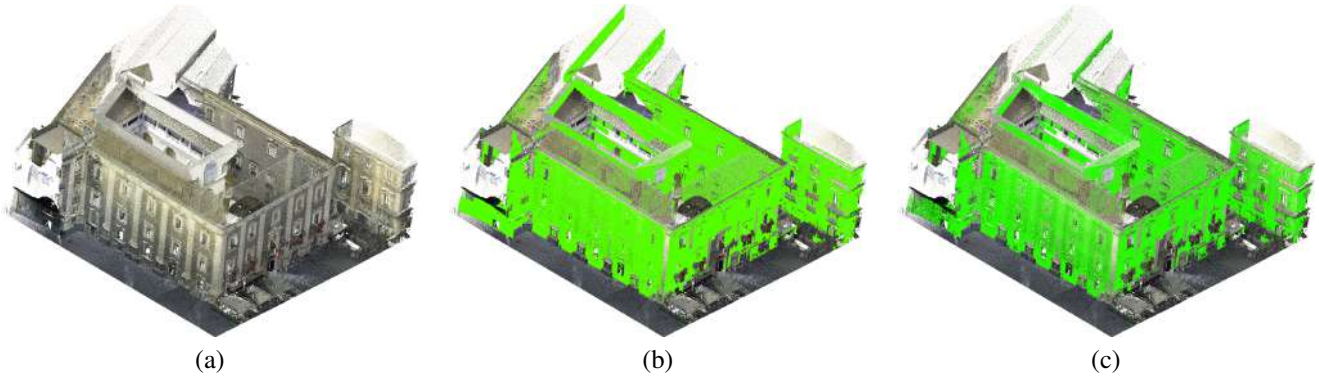


Figure 10: Terrestrial acquisition of an historical building of Catania: in (a) the original point cloud, in (b) the recognized façades are overlaid to the original point cloud and in (c) the points corresponding to façades are downsampled by exploiting the parametric representation of planes.

strict footprint adherence by incorporating context-aware buffer zones, and (ii) a primitive recognition stage based on the Hough Transform, capable of detecting both planar and curved façade structures.

The proposed partitioning approach addresses limitations of traditional footprint-based segmentation, which often neglects contextual geometry and results in misclassification at building boundaries. By flexibly defining spatial regions that include neighboring structures, the method ensures a more reliable grouping of façade-related points.

Evaluation on the H3D dataset demonstrates that our pipeline achieves high recall and produces semantically meaningful façade detections, even in the presence of noisy data. While the measured precision is affected by differences between our façade definition and the one used in the ground truth, the analysis confirms that our method delivers accurate and structured façade interpretations aligned with our semantic criterion.

Looking ahead, several directions can further enhance this work. One line of development focuses on the refinement of buffer zone estimation, with the aim of further improving the segmentation of adjacent buildings and reducing ambiguities at their boundaries. Another promising direction involves the integration of multi-scale features to better capture fine-grained architectural details, such as pilasters or cornices, which can be crucial in certain applications. Moreover, we plan to extend the pipeline to recognize other urban elements beyond façades, such as roofs and courtyards, and other structural features that are important for a complete understanding of the urban fabric.

Finally, an important line of future work will explore how the information extracted by our façade recognition system can be

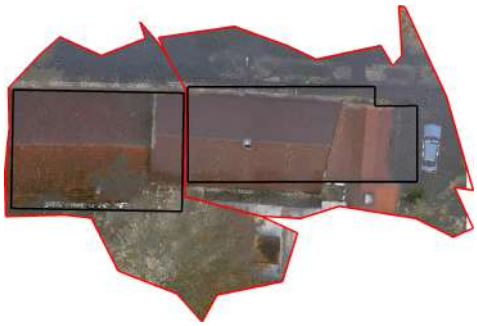


Figure 11: Example of building footprint misalignment leading to the erroneous splitting of a single building into multiple regions. The building identified by the footprint reference of Chunk 49 (right) is partially associated with Chunk 55 (left).

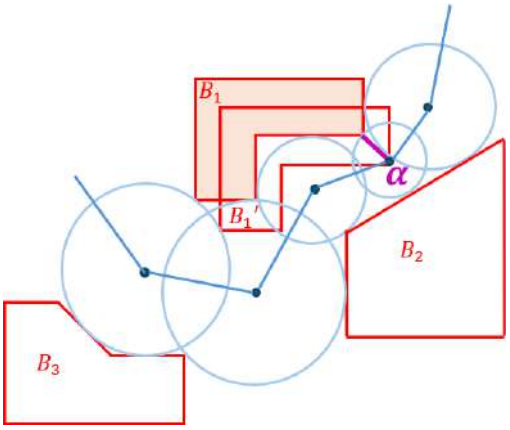


Figure 12: Example of footprint misplacement: building B1 is actually located in the position of B1'; that misplacement is within the buffer zone and does not cause the loss of points during segmentation.

used to refine or correct existing building footprints. By feeding back 3D structural information into 2D cadastral data, we aim to bridge the gap between raw geometric data and structured cartographic representations, thus contributing to the continuous improvement of urban digital twins.

Acknowledgments

All authors are members of the RAISE Innovation Ecosystem, funded by the European Union - NextGenerationEU and by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.5, project "RAISE - Robotics and AI for Socio-economic Empowerment" (ECS00000035).

The authors also acknowledge support from the project PON METRO – ASSE 1 – Agenda Digitale, titled Urban Intelligence Science Hub (UISH) for City Network (CUP: B51B21000430001), funded under the Italian National Operational Program for Metropolitan Cities 2014–2020.

The authors also thank the municipalities of Genova and Catania for the LiDAR data of their cities.

Figure 13: Top view of the point cloud of Chunk 28 resulting from the partitioning strategy. Due to the proximity of footprints, four adjacent buildings are grouped together without explicit separation.

References

[1] American Society for Photogrammetry and Remote Sensing (ASPRS), LAS Specification (ver. 1.4–R15, July 2019), bethesda, Md., American Society for Photogrammetry and Remote Sensing, 27 p. (Jul. 2011).

[2] O. G. Consortium, Citygml 3.0: An opengis® implementation standard, accessed: 2025-02-16 (2020).
URL https://www.opengis.net/doc/IS/CityGML-1/3.0

[3] G. Gröger, L. Plümer, CityGML–interoperable semantic 3D city models, ISPRS Journal of Photogrammetry and Remote Sensing 71 (2012) 12–33. doi:10.1016/j.isprsjprs.2012.04.004.

[4] F. Biljecki, H. Ledoux, J. Stoter, An improved lod specification for 3d building models, Computers Environment and Urban Systems 59 (2016) 25–37. doi:10.1016/j.compenvurbsys.2016.04.005.

[5] F. Lafarge, C. Mallet, Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation, International Journal of Computer Vision 99 (2012) 69–85. doi:10.1007/s11263-012-0517-8.

[6] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, 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), AAAI Press, Portland, OR, USA, 1996, pp. 226–231.

[7] F. Tarsha-Kurdi, T. Landes, P. Grussenmeyer, Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (3/W49A) (2007) 407–412.

[8] J. Elseberg, D. Borrmann, A. Nüchter, One billion points in the cloud – an octree for efficient processing of 3d laser scans, ISPRS Journal of Photogrammetry and Remote Sensing 76 (2013) 76–88. doi:10.1016/j.isprsjprs.2012.10.004.

[9] J. Papon, A. Abramov, M. Schoeler, F. Wörgötter, Voxel cloud connectivity segmentation - supervoxels for point clouds, in: CVPR, 2013, pp. 2027–2034.

[10] Z. Su, J. Peng, D. Feng, S. Li, Y. Yuan, G. Zhou, A building point cloud extraction algorithm in complex scenes, Remote Sensing 16 (11). doi:10.3390/rs16111934.

[11] C. R. Qi, H. Su, K. Mo, L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: CVPR, 2017, pp. 652–660.

[12] L. Landrieu, M. Simonovsky, Large-scale point cloud semantic segmentation with superpoint graphs, CVPR.

[13] A. Kaiser, J. A. Ybanez Zepeda, T. Boubekeur, A Survey of Simple Geometric Primitives Detection Methods for Captured 3D Data, Computer Graphics Forum 38 (1) (2019) 167–196. doi:10.1111/cgf.13451.

[14] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for Point-Cloud Shape Detection, Computer Graphics Forum 26 (2) (2007) 214–226.

[15] F. A. Limberger, M. M. Oliveira, Real-time detection of planar regions in unorganized point clouds, Pattern Recognition 48 (6) (2015) 2043 – 2053. doi:10.1016/j.patcog.2014.12.020.

[16] M. Attene, G. Patanè, Hierarchical structure recovery of point-sampled surfaces, Computer Graphics Forum 29 (6) (2010) 1905–1920. doi:10.1111/j.1467-8659.2010.01658.x.

[17] T. Le, Y. Duan, A primitive-based 3D segmentation algorithm for mechanical CAD models, Computer Aided Geometric Design 52-53 (2017) 231–246, Geometric Modeling and Processing 2017. doi:10.1016/j.cagd.2017.02.009.

[18] L. Li, M. Sung, A. Dubrovina, L. Yi, L. J. Guibas, Supervised Fitting of Geometric Primitives to 3D Point Clouds, in: Proceedings – IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019, pp. 2652–2660. doi:10.1109/CVPR.2019.00276.

[19] M. A. Uy, Y.-Y. Chang, M. Sung, P. Goel, J. G. Lambourne, T. Birdal, L. J. Guibas, Point2cyl: Reverse engineering 3d objects from point clouds to extrusion cylinders, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11850–11860.

[20] Y. Qie, L. Qiao, N. Anwer, Enhanced invariance class partitioning using discrete curvatures and conformal geometry, Computer-Aided Design 133. doi:10.1016/j.cad.2020.102985.

[21] F. Poux, C. Mattes, Z. Selman, L. Kobbelt, Automatic region-growing system for the segmentation of large point clouds, Automation in Construction 138 (2022) 104250. doi:10.1016/j.autcon.2022.104250.

[22] V. Markovic, Z. Jakovljevic, I. Budak, Automatic recognition of cylinders and planes from unstructured point clouds, The Visual Computer (2021) 1–24.

[23] T. Birdal, B. Busam, N. Navab, S. Ilic, P. Sturm, Generic primitive detection in point clouds using novel minimal quadric fits, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (6) (2020) 1333–1347. doi:10.1109/TPAMI.2019.2900309.

[24] C. Liu, W. Hu, Real-time geometric fitting and pose estimation for surface of revolution, Pattern Recognition 85 (2019) 90–108. doi:10.1016/j.patcog.2018.08.002.

[25] F. Bergamasco, M. Pistellato, A. Albarelli, A. Torsello, Cylinders extraction in non-oriented point clouds as a clustering problem, Pattern Recognition 107 (2020) 107443. doi:10.1016/j.patcog.2020.107443.

[26] G. Sharma, D. Liu, S. Maji, E. Kalogerakis, S. Chaudhuri, R. Měch, ParSeNet: A Parametric Surface Fitting Network for 3D Point Clouds, in: Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII, Springer-Verlag, Berlin, Heidelberg, 2020, p. 261–276. doi:10.1007/978-3-030-58571-6_16.

[27] S. Yan, Z. Yang, C. Ma, H. Huang, E. Vouga, Q. Huang, HPNet: Deep Primitive Segmentation Using Hybrid Representations, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 2753–2762. doi:10.1109/ICCV48922.2021.00275.

[28] G. Sharma, B. Dash, A. RoyChowdhury, M. Gadelha, M. Loizou, L. Cao, R. Wang, E. G. Learned-Miller, S. Maji, E. Kalogerakis, PriFit: Learning to Fit Primitives Improves Few Shot Point Cloud Segmentation, Computer Graphics Forumdoi:10.1111/cgf.14601.

[29] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, Randla-net: Efficient semantic segmentation of large-scale point clouds, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11108–11117.

[30] S. González-Collazo, N. Canedo-González, E. González, J. Balado, Semantic point cloud segmentation in urban environments with 1d convolutional neural networks, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 48 (2024) 205–211.

[31] T. Hackel, J. D. Wegner, K. Schindler, Fast semantic segmentation of 3d point clouds with strongly varying density, ISPRS annals of the photogrammetry, remote sensing and spatial information sciences 3 (2016) 177–184.

[32] X. Sun, B. Guo, C. Li, N. Sun, Y. Wang, Y. Yao, Semantic segmentation and roof reconstruction of urban buildings based on lidar point clouds, ISPRS International Journal of Geo-Information 13 (1) (2024) 19.

[33] E. K. Dey, M. Awrangjeb, F. Tarsha Kurdi, B. Stantic, Machine learning-based segmentation of aerial lidar point cloud data on building roof, European Journal of Remote Sensing 56 (1) (2023) 2210745.

[34] L. Li, J. Yao, J. Tu, X. Liu, Y. Li, L. Guo, Roof plane segmentation from airborne lidar data using hierarchical clustering and boundary relabeling, Remote Sensing 12 (9) (2020) 1363.

[35] X. Wang, S. Ji, Roof plane segmentation from lidar point cloud data using region expansion based l 0 gradient minimization and graph cut, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14 (2021) 10101–10116.

[36] Z. Li, J. Shan, RANSAC-based multi primitive building reconstruction from 3D point clouds, ISPRS Journal of Photogrammetry and Remote Sensing 185 (2022) 247–260.

[37] J. Niemeyer, F. Rottensteiner, U. Sörgel, C. Heipke, Hierarchical higher order crf for the classification of airborne lidar point clouds in urban areas, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives 41 (2016) 41 (2016) 655–662.

[38] R. Zhao, M. Pang, J. Wang, Classifying airborne lidar point clouds via deep features learned by a multi-scale convolutional neural network, International journal of geographical information science 32 (5) (2018) 960–979.

[39] S. Xia, D. Chen, R. Wang, J. Li, X. Zhang, Geometric primitives in lidar point clouds: A review, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 13 (2020) 685–707. doi:10.1109/JSTARS.2020.2969119.

[40] A. Klimkowska, S. Cavazzi, R. Leach, S. Grebby, Detailed three-dimensional building favbade reconstruction: A review on applications, data and technologies, Remote Sensing 14 (11). doi:10.3390/rs14112579.

[41] H. Xu, C.-Y. Chen, 3d facade reconstruction using the fusion of images and lidar: A review, in: International Computer Symposium, Springer, 2018, pp. 178–185.

[42] J. Wang, Y. Xu, O. Remil, X. Xie, N. Ye, C. Yi, M. Wei, Automatic modeling of urban facades from raw lidar point data, in: Computer Graphics Forum, Vol. 35, Wiley Online Library, 2016, pp. 269–278.

[43] T. Kelly, J. Femiani, P. Wonka, N. J. Mitra, Bigsur: large-scale structured urban reconstruction, ACM Trans. Graph. 36 (6). doi:10.1145/3130800.3130823.

[44] L. Nan, C. Jiang, B. Ghanem, P. Wonka, Template assembly for detailed urban reconstruction, Computer Graphics Forum 34 (2) (2015) 217–228. doi:10.1111/cgf.12554.

[45] J. Xiao, M. Gerke, G. Vosselman, Building extraction from oblique airborne imagery based on robust façade detection, ISPRS Journal of Photogrammetry and Remote Sensing 68 (2012) 56–68. doi:10.1016/j.isprsjprs.2011.12.006.

[46] W. Hao, Y. Wang, W. L. and, Slice-based building facade reconstruction from 3d point clouds, International Journal of Remote Sensing 39 (20) (2018) 6587–6606. doi:10.1080/01431161.2018.1463113.

[47] L. Truong-Hong, D. F. Laefer, Octree-based, automatic building facade generation from lidar data, Computer-Aided Design 53 (2014) 46–61.

[48] Y. Wang, Y. Ma, A. xing Zhu, H. Zhao, L. Liao, Accurate facade feature extraction method for buildings from three-dimensional point cloud data considering structural information, ISPRS Journal of Photogrammetry and Remote Sensing 139 (2018) 146–153. doi:10.1016/j.isprsjprs.2017.11.015.

[49] X. Hu, Y. Wang, T. Zhou, D. S. M. and, Building facade structure extraction method based on three-dimensional laser point cloud by considering semantic information, Annals of GIS 30 (3) (2024) 291–305. doi:10.1080/19475683.2024.2335953.

[50] F. Wang, G. Zhou, H. Hu, Y. Wang, B. Fu, S. Li, J. Xie, Reconstruction of lod-2 building models guided by façade structures from oblique photogrammetric point cloud, Remote Sensing 15 (2) (2023) 400.

[51] B. Pantoja-Rosero, R. Achanta, M. Kozinski, P. Fua, F. Perez-Cruz, K. Beyer, Generating LOD3 building models from structure-from-motion and semantic segmentation, Automation in Construction 141 (2022) 104430.

[52] A. Duran, C. Waibel, V. Piccioni, B. Bickel, A. Schlueter, A review on artificial intelligence applications for facades, Building and Environment 269 (2025) 112310. doi:10.1016/j.buildenv.2024.112310.

[53] C. Romanengo, D. Cabiddu, S. Pittaluga, M. Mortara, Building semantic segmentation from large-scale point clouds via primitive recognition, Graphical Models 136 (10123) (2024) 4.

[54] F. Pirotti, C. Zanchetta, M. Previtali, S. Della Torre, Detection of building roofs and facades from aerial laser scanning data using deep learning, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W11 (2019) 975–980. doi:10.5194/isprs-archives-XLII-2-W11-975-2019.

[55] M. Weinmann, M. Weinmann, C. Mallet, M. Brédif, A classification-segmentation framework for the detection of individual trees in dense mms point cloud data acquired in urban areas, Remote Sensing 9 (3).

doi:10.3390/rs9030277.
[56] K. Hormann, A. Agathos, The point in polygon problem for arbitrary polygons, Computational Geometry 20 (3) (2001) 131–144. doi:10.1016/S0925-7721(01)00012-8.
[57] C. Jordan, Cours d'analyse de l'École polytechnique, Vol. 1, Gauthier-Villars et fils, 1893.
[58] W. R. Franklin, Pnpoly - point inclusion in polygon test, last update: March 2023 (1970).
URL https://wrfranklin.org/Research/Short_Notes/pnpoly.html
[59] The CGAL Project, CGAL User and Reference Manual, 6.0.1 Edition, CGAL Editorial Board, 2024.
URL https://doc.cgal.org/6.0.1/Manual/packages.html
[60] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, in: Proceedings of the 19th annual conference on computer graphics and interactive techniques, 1992, pp. 71–78.
[61] P. Mukhopadhyay, B. B. Chaudhuri, A survey of Hough transform, Pattern Recognition 48 (3) (2015) 993 – 1010. doi:10.1016/j.patcog.2014.08.027.
[62] P. V. C. Hough, Method and means for recognizing complex patterns, US Patent 3,069,654 (1962).
[63] M. C. Beltrametti, L. Robbiano, An algebraic approach to Hough transforms, Journal of Algebra 37 (2012) 669–681.
[64] A. Raffo, C. Romanengo, B. Falcidieno, S. Biasotti, Fitting and recognition of geometric primitives in segmented 3d point clouds using a localized voting procedure, Computer Aided Geometric Design 97 (2022) 102123. doi:10.1016/j.cagd.2022.102123.
[65] M. Livesu, cinolib: a generic programming header only c++ library for processing polygonal and polyhedral meshes, Transactions on Computational Science XXXIVHttps://github.com/mlivesu/cinolib/. doi:10.1007/978-3-662-59958-7_4.
[66] GDAL/OGR contributors, GDAL/OGR Geospatial Data Abstraction software Library, Open Source Geospatial Foundation (2025). doi:10.5281/zenodo.5884351.
URL https://gdal.org
[67] M. Kölle, D. Laupheimer, S. Schmohl, N. Haala, F. Rottensteiner, J. D. Wegner, H. Ledoux, 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 Journal of Photogrammetry and Remote Sensing 1 (2021) 100001. doi:10.1016/j.ophoto.2021.100001.
[68] OpenStreetMap contributors, Planet dump retrieved from https://planet.osm.org, https://www.openstreetmap.org, accessed: 2024-11-01 (2024).
[69] Comune di Genova, Mapstore2 3d city model viewer, Web Application, interactive 3D viewer for CityGML data of Genoa, Italy (2023).
URL https://mappe.comune.genova.it/MapStore2/#/viewer/1000003072