

SimpliCity: Reconstructing Buildings with Simple Regularized 3D Models

Jean-Philippe Bauchet^{1*}, Raphael Sulzer^{1,2*}, Florent Lafarge², Yuliya Tarabalka¹

¹LuxCarta Technology, Mouans-Sartoux, France

²Centre INRIA d’Université Côte d’Azur, Sophia Antipolis, France

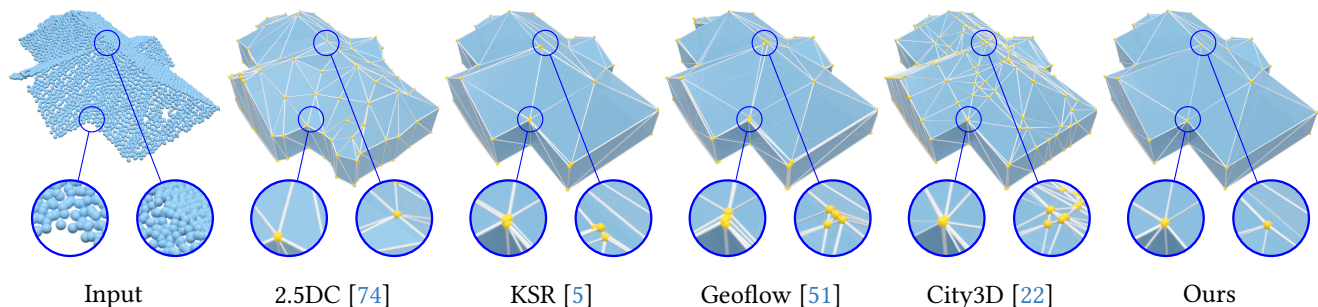


Figure 1. **SimpliCity**. Our building reconstruction method produces a simple, regularized mesh while being a faithful approximation of the input Lidar scan. In contrast, the output meshes of Geoflow [51] and City3D [22] contain a much higher number of facets, including tiny ones that correct connectivity approximations around corners.

Abstract

Automatic methods for reconstructing buildings from airborne LiDAR point clouds focus on producing accurate 3D models in a fast and scalable manner, but they overlook the problem of delivering simple and regularized models to practitioners. As a result, output meshes often suffer from connectivity approximations around corners with either the presence of multiple vertices and tiny facets, or the necessity to break the planarity constraint on roof sections and facade components. We propose a 2D planimetric arrangement-based framework to address this problem. We first regularize, not the 3D planes as commonly done in the literature, but a 2D polyhedral partition constructed from the planes. Second, we extrude this partition to 3D by an optimization process that guarantees the planarity of the roof sections as well as the preservation of the vertical discontinuities and horizontal rooftop edges. We show the benefits of our approach against existing methods by producing simpler 3D models while offering a similar fidelity and efficiency.

1. Introduction

Reconstructing buildings in 3D from physical data measurements is a long standing problem within the computer vision, remote sensing and computer graphics communities [40, 62]. While dense mesh-based representations are commonly used for visualization and immersive tasks, e.g. GoogleEarth [19], lightweight 3D models of buildings are often required in many large-scale application scenarios such as simulation of physical phenomena, urban planning or navigation [6, 14]. In these scenarios, a building is represented by a surface mesh with a low number of facets, ideally just enough to describe its structure.

Among possible data sources, airborne Lidar scanning offers several advantages. It directly produces 3D point clouds with a high precision and a density that can reach up to 50 points per squared meter. The emitted pulses can also provide descriptions of buildings that are occluded by vegetation or shadows in optical imagery, making them hard to detect and reconstruct. Airborne Lidar scans used to have limited availability, but in a renewed interest for the construction of a digital twin, public organizations, such as the French National Mapping Agency [13] or the Swiss Federal Office of Topography [47] have acquired and released Lidar data at a country scale. In addition, this public

*Both authors contributed equally to the paper.

Lidar data is often already semantically classified [17].

We address the problem of reconstructing lightweight building models from airborne Lidar point clouds with the following objectives:

- **Fidelity:** output meshes should constitute of a faithful approximation of the data,
- **Simplicity:** output meshes should be composed of a low number of facets and preserve the geometric regularities of buildings, if any,
- **Efficiency:** the algorithm should be fast, scalable, and proceed with a low number of user parameters,
- **Geometric guarantees:** the output meshes should be watertight, 2d-manifold, intersection-free, and should conform to the CityGML LOD2 level of detail [20] with, in particular, the planarity of facets composing a facade component or a roof section.

Unfortunately, existing methods do not perform well on these four objectives simultaneously. Robust approaches [4, 22, 71, 76] usually target high fidelity and efficiency by detecting planes and assembling them either directly in 3D or within a planimetric arrangement extruded to 3D. Although planes can be accurately and efficiently detected from airborne Lidar, plane-based approaches often generate output meshes that lack regularity, simplicity and important geometric guarantees. A recurring problem is that four or more planes are unlikely to be connected in exactly one vertex leading to the creation of tiny facets approximating the one-vertex connection, or the loss of the planarity property of polygonal facets (*cf.* Fig. 1). Furthermore, geometric regularities are often poorly preserved in the output models. Regularizing detected planes [58, 76] can help but does not guarantee a regular output mesh free of geometric defects.

In this work, we propose a 2D planimetric arrangement-based framework to address these issues. The two key ideas are first to regularize, not the 3D planes as in the literature, but the 2D polyhedral partition constructed from the planes, and second to extrude in 3D this partition by an optimization process that guarantees the planarity of the roof sections as well as the preservation of the vertical discontinuities and horizontal rooftop edges. Our solution is efficient in solving issues (i) and (ii) as the 2D polygonal partition exhibits a simple facet connectivity while being highly regularized. We show the benefits of our approach against prior methods by producing more simple 3D models while reaching a similar fidelity and efficiency.

2. Related work

Our review of previous work discusses the main strategies for reconstructing buildings from input point clouds. Note that a vast literature also exists from other data sources such as single-view satellite image [37, 38, 73] or dense meshes generated by multiview stereo pipelines [21, 77].

Model-driven. These methods typically perform template matching from a predefined library of building parts [25, 43, 59]. Roof topology graphs [65, 66] or rules derived from constructive solid geometry [28] can be used to guide the matching. Such methods produce realistic results, but lose generality when applied to urban landscapes that cannot be precisely described using the predefined templates. The Manhattan-world assumption can also be used to restrict the orientations of buildings to three orthogonal directions, and reconstruct them as polycubes [31, 57].

Mesh simplification. Other methods simplify a dense mesh reconstructed from the input point cloud [7, 26, 56] into a more concise mesh. A common strategy consists in iteratively collapsing edges depending on quadric error metrics [16]. To better preserve the structure of piece-wise-planar objects like buildings, these edge contraction operators may consider planar proxies detected during a preprocessing stage [30, 54, 61]. Closely related to these simplification techniques, dual contouring meshing techniques can also be used for reconstruct buildings with 2.5D-view dependent triangle meshes [74, 75].

Plane assembly. Detecting planes from the input data and connecting them into a mesh is also a popular strategy.

Plane detection is traditionally performed using Hough transform [50], RANSAC [55] or region-growing procedures [27, 53], all these methods requiring the tuning of a few parameters. Neural architectures [24, 29, 67, 72], which are parameter-free, show promising results. Trained on synthetic CAD databases, they however do not generalize well on real-world data. Yu *et al.* [70] propose an energy-based model in which the retrieved planes are iteratively refined to satisfy coverage or compactness conditions.

Subsequent plane assembling techniques then either construct a connectivity graph between the detected planes [8, 11, 69] to find out the vertices and edges composing the output mesh, or compute a space decomposition by plane slicing operations before extracting a mesh from it [5, 9, 23, 41]. In particular, PolyFit [41] has been used in several building reconstruction pipelines [22, 35, 64]. However, due to the complexity of its binary linear optimization, these approaches do not scale well to complex structures composed of a few dozens or hundred planar components. The space decomposition can also be done in 2D through a planimetric arrangement which is then extruded to 3D [10, 27, 32, 51, 52]. These solutions offer a good accuracy but often produce overly-complex output meshes, *e.g.* with tiny facets that adjust the connection between four planes or more.

Neural models. Plane assembly based methods can also be combined with a learned occupancy field [12]. Such methods require a minimal number of points for verti-

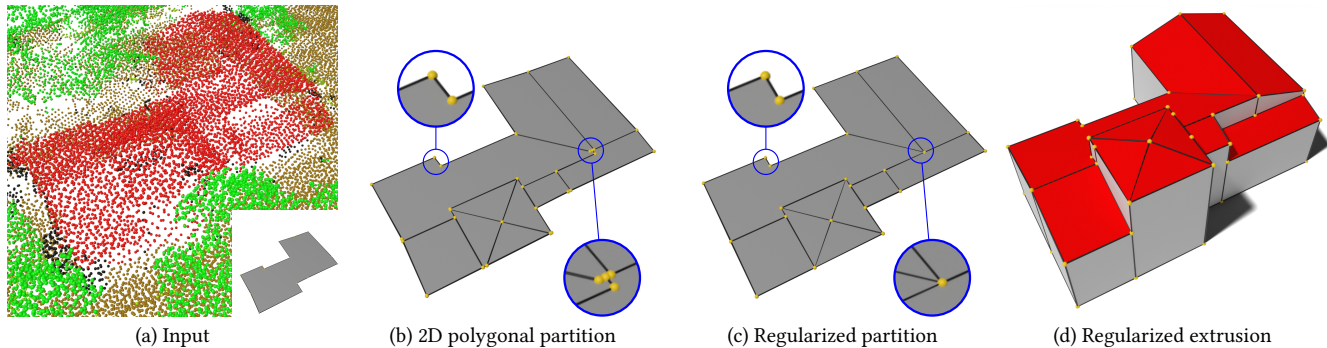


Figure 2. **Overview.** Starting from a LiDAR scan and a building footprint (a), we first construct a 2D polygonal partition of the roof structure (b), Section 3.2, Fig. 3). The partition is then regularized to both enforce orthogonality, parallelism and collinearity between edges and simplify the vertex layout (c), Section 3.3, Fig. 4). Finally, the partition is extruded using an optimization procedure that preserves the planarity of roof sections and the horizontality of rooftop edges (d), Section 3.4, Fig. 5).

cal surfaces to deliver accurate results, which is hard to guarantee in practice. Furthermore, the learned occupancy field does not generalize well to unknown complex building types. Liu *et al.* [36] propose a generative model that predicts sequences of vertices and faces that form the output polygonal mesh. However, the output does not adhere to strong geometric guarantees.

Building regularization. Reconstructing 3D models that preserve geometric regularities contained in the building structures can be addressed by plane assembly methods. One strategy consists in regularizing configurations of planes either after their detection [34, 45] or during [39, 44, 70]. Exploited in [33, 58, 76] with roof symmetry and facade orthogonality and parallelism, this solution helps but does not guarantee highly-regular output meshes. This can also be done in 2D with line-segments [2, 3] or polygons [22] describing the building contours. To our knowledge, only Vuillamy *et al.* [60] try to regularize a space partition directly. They simplify 2D polygonal partitions by encouraging line concurrency and orthogonality only. Moreover, their point-line projective duality formulation requires high computing resources and does not scale to complex partitions.

3. Proposed method

3.1. Overview

Our algorithm takes as input (i) an airborne Lidar point cloud, and (ii) a set of polygonal footprints describing the contours of buildings. The latter can be either obtained via online cadaster map databases, *e.g.* [49], or computed by an automatic building contouring method [1, 4, 68].

The output 3D models are polygon surface meshes that are, by construction, watertight, 2-manifold and intersection-free. The polygonal facets can be decomposed

into triangles using a constrained Delaunay triangulation with the guarantee that triangles from the same polygonal facet are exactly co-planar. This triangle decomposition is used later in Section 4 to fairly compare the mesh complexity with competitors.

Fig. 2 illustrates the three main steps of our algorithm, namely (i) the construction of 2D polygonal partitions describing the roof structures, (ii) the regularization and simplification of these partitions, and (iii) the extrusion of the partitions to 3D.

3.2. Construction of 2D polygonal partitions

The first step consists in constructing a 2D polygonal partition that represents a projection of the roof structure to the horizontal plane. This partition is not purely geometric, but is enriched by 3D information. Each polygonal cell of the partition is associated with a 3D plane approximating the corresponding roof section in 3D space. We start by detecting 3D planes from the input point cloud. The detection is controlled by two parameters: a fitting tolerance that specifies the maximal distance of an inlier point to its associated plane, and a minimal number of inliers per plane that avoids the detection of too small components. Because airborne laser scans are usually acquired at a near-nadir angle they include only little information on vertical components of the building. Therefore, the detected planes mostly correspond to non-vertical roof sections of the building. To also recover vertical sections we continue as follows. For each detected plane, we compute a 3D planar alpha shape [15] as the 2D alpha shape of a set of inlier points projected into the plane, later referred to as a planar primitive. We now extract two types of 3D line segments from the configuration of planar primitives. The first type corresponds to a potential connection between two adjacent roof sections. These line segments, referred as intersection lines, are computed as the intersections between the pairs of adjacent

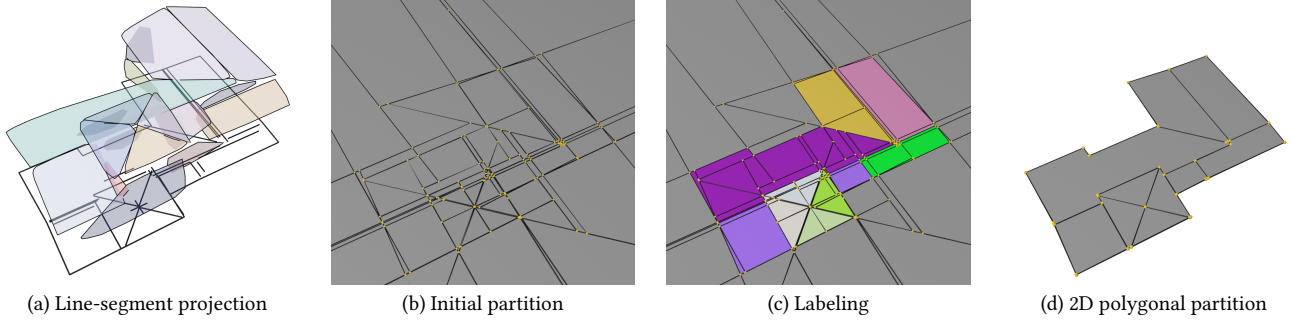


Figure 3. **Construction of 2D polygonal partition.** Line-segments at the intersection of adjacent detected planes and on their boundary are projected into the horizontal plane, jointly with the input footprint line-segments (a). An initial, dense 2D polygonal partition is built from all these line-segments by kinetic simulation (b) and enriched by plane labels ((c), colored polygons). Polygonal cells with same label are then regrouped to form a 2D polygonal partition that describes the roof structure (d).

planar primitives. The second type, referred as discontinuity lines, corresponds to vertical discontinuities in the roof structure. These line segments are a subset of edges composing the planar primitive contours after simplification [42]. We use all edges whose difference of the average heights of the points located on both sides is greater than a given threshold (set to 50 cm in our experiments). The sets of intersection and discontinuity lines are then projected to the horizontal plane together with the lines forming the input polygonal footprint (Fig. 3a). We then extend all the line segments at constant speed within a kinetic simulation [3] to form an initial 2D polygonal partition (Fig. 3b).

Next, we assign a label to each cell of the partition that corresponds to one of the detected planes or to the ground. This assignment procedure is formulated as an energy minimization problem with discrete variables. Let $C = (c_1, c_2 \dots c_N)$ be a polygonal partition with N cells, and $L = \{0, 1 \dots M\}$ be a vector of plane indices, where M is the number of planar primitives initially extracted. Each cell c_i defines a discrete variable $x_i \in L$. $x_i = 0$ means that the related cell is not assigned to any of the planes, which can be the case if c_i is outside the building footprint, for instance. We obtain an optimal label assignment $X^* = (x_1, x_2 \dots x_N)$ by minimizing the energy

$$E(X) = E_d(X) + E_p(X) + E_c(X) \quad (1)$$

where $E_d(X)$ is a data term encouraging a variable x_i to accept a label that corresponds to a planar primitive that is next to the cell c_i . $E_p(X)$ is a pairwise smoothness term penalizing a label difference for adjacent cells c_i and c_j , based on the height difference on the edge that is common to those cells. Note that this definition does not penalize a difference of labels in ridge and hip lines, where two planes intersect in the 3D space. $E_c(X)$ is a complexity term, set to the total number of edges required, to describe the M polygons resulting from the fusion of all cells with same labels. Because $E(X)$ is not convex and lives

in a discrete domain, its optimization is a difficult problem. We search for an approximate solution using an iterative scheme. Given an initial label assignment X_0 , we generate a set of neighbor configurations, in which one or several cells from the partition, are transferred from one current roof section to another. Configurations associated with an energy decrease are sorted in a priority queue. Then, we pop from the queue the configuration that corresponds to the largest energy drop, and repeat the process until the queue gets empty. Each cell of the initial partition with a cell label other than ground now corresponds to a roof section and is used as a final polygon in the 2D polygonal partition (Fig. 3d).

3.3. Regularization of 2D polygonal partitions

The second step of our algorithm aims to simplify the 2D polygonal partition and enhance its regularity. We address this problem with a global optimization procedure under geometric constraints.

Many building roof types include corner types that require four or more planar components to meet in the same point. However, 3D planes detected from Lidar point clouds do not exhibit such a behaviour. Fig. 4a shows how this problem creates an overly complex polygonal partition, leading to the presence of extra vertices and small facets in the final output mesh. To solve this issue, we start by collapsing short edges with a length smaller τ_h in the 2D polygonal partition (Fig. 4b). To enhance the regularity of the partition we detect near parallel and orthogonal edges. We then build a graph G , where each node represents an edge of the partition, and an edge in G represents a near-parallel or near-orthogonal pair of edges in the partition (Fig. 4c). In practice, we compute G before collapsing short edges, as the edge collapse can significantly alter edge orientations. We finally apply a global optimization procedure to find new positions for all vertices of the partition that respect the parallel and orthogonal constraints of

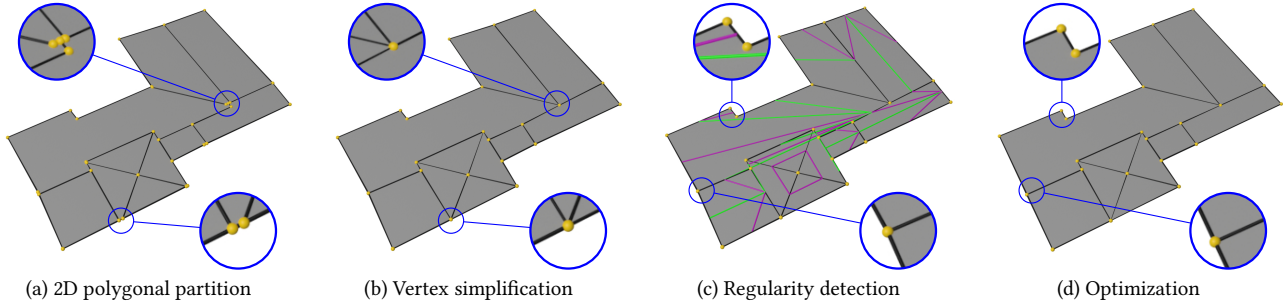


Figure 4. **Regularization of 2D polygonal partition.** The 2D polygonal partition (a) is first simplified by collapsing small edges (b), see close-ups). The regularity graph G is then built by detecting pairs of near-parallel edges (green lines) and near-orthogonal edges (purple lines) in (c). A global optimization of the vertex coordinates constrained by G is then performed to regularize the partition (d).

adjacent edges (Fig. 4d). Note that the 3D plane equation associated with each cell of the initial partition is preserved by these geometric operations.

3.4. Extrusion

Once regularized, the 2D polygonal partition is extruded by projecting each cell to 3D using the associated plane equations. One problem of such an extrusion strategy is that two adjacent roof sections, e.g. the two sides of a gabled roof, are not modeled with a continuous transition (Fig. 5b). This is because the same vertex in the 2D partition can be extruded to slightly different height values from the different plane equations associated with its incident cells. Simply extruding each vertex to the mean height value of all incident cells breaks the planarity of the reconstructed roof sections (Fig. 5c). To force transition continuity between close adjacent roof sections without breaking their planarity, we propose a global optimization procedure under geometric constraints, similar to the partition regularization problem in Section 3.3. We first collapse 3D vertices with a height distance smaller τ_v and then optimize the height value of each vertex with the constraint that all vertices per roof section need to adhere to the same plane equation (Fig. 5d). Note that, we keep the x- and y-coordinates of the vertices fixed to not break the parallel and orthogonal symmetries enforced during the horizontal optimization. After horizontal and vertical optimization we finally assemble the obtained roof polygonal facets and vertical walls to form the 2.5D building model.

4. Experiments

4.1. Experimental setup

Dataset. We evaluate our method on three different datasets:

- The full test set of the *Building3D Tallinn dataset* [63], comprising 3472 building models, at an approximate

density of 27 points per squared meter;

- The *Zurich dataset* [47] with Lidar point clouds sampled at approximately 12 points per squared meter, for a total area of 4 km^2 , for which 3177 reference building models [48] are provided in a CityGML LOD2.3 format;
- The *Helsinki dataset* [46] with Lidar point clouds sampled at a density of 56 points per squared meter, covering 4 km^2 in total, for which 1275 reference building models are provided in a CityGML LOD2.2 format.

Metrics. We measure the quality of building models using the following metrics. For complexity, we count the number of vertices $|V|$ and facets $|F|$ of the model, as well as the ratio $E_{<0.5 \text{ m}}$ of edges lower than 0.5 meter. For accuracy, we sample 100k points on the reconstructed models as well as the reference models and measure (i) the one-sided Chamfer distance $\text{CD}_{\text{Rec} \rightarrow \text{Ref}}$ from reconstruction to reference samples, and (ii) the one-sided Chamfer distance $\text{CD}_{\text{Imp} \rightarrow \text{Rec}}$ from the input point cloud to the reconstruction samples. We do not measure the distance from reference to reconstructed models because we find that some of the reference models have interior facets that produce irrelevant distance measures.

Baselines. We compare our method with four different baselines. Most of our competitors rely on the extraction of planar shapes from the input point clouds. Though more sophisticated techniques could be used, we extract planes from point clouds using a standard region-growing procedure [27], so that the same configuration of planes can be input to our method and all baselines.

- **2.5D Dual Contouring (2.5DC)** [74] detects sharp feature points from the input point cloud and triangulates these points to form 2.5D semi-dense mesh models. Vertices of the mesh are then snapped to detected principal directions for regularization. It takes a building point

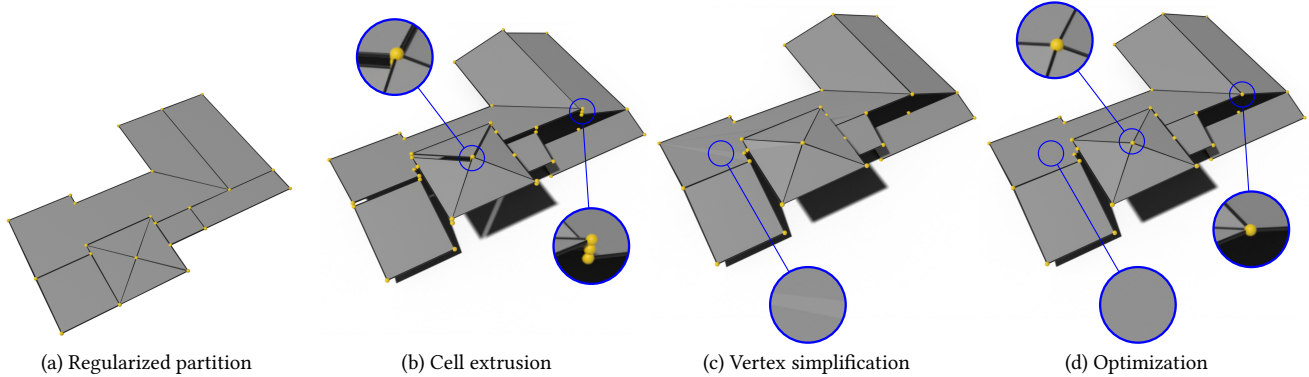


Figure 5. **Extrusion.** Each cell of the 2D regularized partition (a) is first extruded according to its associated detected plane (b). Close 3D vertices with same x- and y-coordinates are then merged, breaking the planarity of the extruded facets (c). The z-coordinates of the vertices are finally readjusted using a global optimization under facet planarity constraints (d).

cloud augmented with normals as input.

- **Kinetic Shape Reconstruction (KSR)** [5] uses planes detected on the input point cloud and computes a polyhedral decomposition. The cells of the decomposition are labelled as inside or outside using point normal orientation. Finally, a building model is extracted as a concise polygon mesh. For a fair comparison, we input the same plane configuration to KSR and our method, including planes for all vertical discontinuities and per line segment of the footprint polygon.
- **Geoflow** [51] is a 2.5D extrusion method that works similar to ours, but without any optimization steps to improve the simplicity of building models. It takes a building point cloud and footprint as input.
- **City3D** [22] is another plane arrangement method based on PolyFit [41]. The pipeline detects planes from the input point cloud and from vertical discontinuities. To target the simplicity of models, the input footprint is regularized before planes are extracted from its line segments.

4.2. Results

Tab. 1 provides the quantitative results of the comparative study while Fig. 6 shows visual results on various buildings.

Our method produces meshes that are significantly simpler than competitors on the three datasets. The average number of vertices of the prior methods is two to four times more than ours. This gap is particularly important with Geoflow [51] which also exploits a planimetric arrangement-based framework like us, but with a naive extrusion and no regularization of the partition. Our meshes also exhibit a much lower ratio of small edges than the competitors, except for 2.5DC [74] that, by construction, displaces hermite vertices, but without guaranteeing the planarity of roof sections and facade components.

The significant gain on complexity metrics does not affect performance as our method is among the fastest be-

Table 1. **Quantitative evaluation.** For each metric and each dataset, we highlight the **best** and **second best** scores. [†]We stop the process after a runtime of 5 min per building.

		Complexity			Accuracy		Perform.
		$ V $	$ F $	$E_{<0.5\text{ m}}$ [%]	$CD_{\text{Inp} \rightarrow \text{Rec}}$ [cm]	$CD_{\text{Rec} \rightarrow \text{Ref}}$ [cm]	Time [s]
Tallinn	Reference [63]	84.6	47.5	1.28	6.68	-	-
	2.5DC [74]	112	187	2.86	11.9	81.9	0.13
	KSR [5]	67.8	71.4	8.35	8.07	15.9	10.4
	Geoflow [51]	180	172	28.0	7.18	36.4	4.03
	City3D [†] [22]	116	199	25.0	9.39	7.24	49.1
	Ours	32.2	58.2	4.39	9.34	13.2	4.19
Zürich	Reference [48]	54.6	77.2	4.48	58.3	-	-
	2.5DC [74]	140	241	2.57	16.4	50.6	0.052
	KSR [5]	116	139	8.72	22.0	33.2	6.52
	Geoflow [51]	227	220	27.0	28.1	50.0	1.59
	City3D [†] [22]	112	193	28.3	30.6	20.0	42.3
	Ours	58.3	109	7.48	26.6	26.0	3.82
Helsinki	Reference [46]	203	236	8.97	66.2	-	-
	2.5DC [74]	570	1045	2.67	22.6	69.9	1.15
	KSR [5]	619	704	14.5	39.8	39.2	67.9
	Geoflow [51]	900	883	23.1	34.2	54.7	6.12
	Ours	214	393	9.5	32.3	37.1	6.81

hind 2.5DC [74] and more than one order magnitude faster than City3D [22]. Notably, reducing complexity also limits expensive downstream operations on the geometric elements forming the output mesh. The gain in regularity moderately impacts accuracy with typically a precision loss of a few centimeters with respect to the most accurate methods. Note that our method exhibits a competitive score on both Chamfer distances (i.e. measured from input points and to reference model), in contrast to 2.5DC [74] or Geoflow [51] that perform well on only one of the two.

Regarding the geometric properties of the reconstructed models, our algorithm offers the most desired guarantees with watertight, 2-manifold, intersection-free meshes. Similarly to 2.5DC [74] and Geoflow [51], our algorithm relies upon a 2.5D-view dependent representation of buildings. Only KSR [5] and City3D [22] can produce full 3D models, but also require higher computing resources.

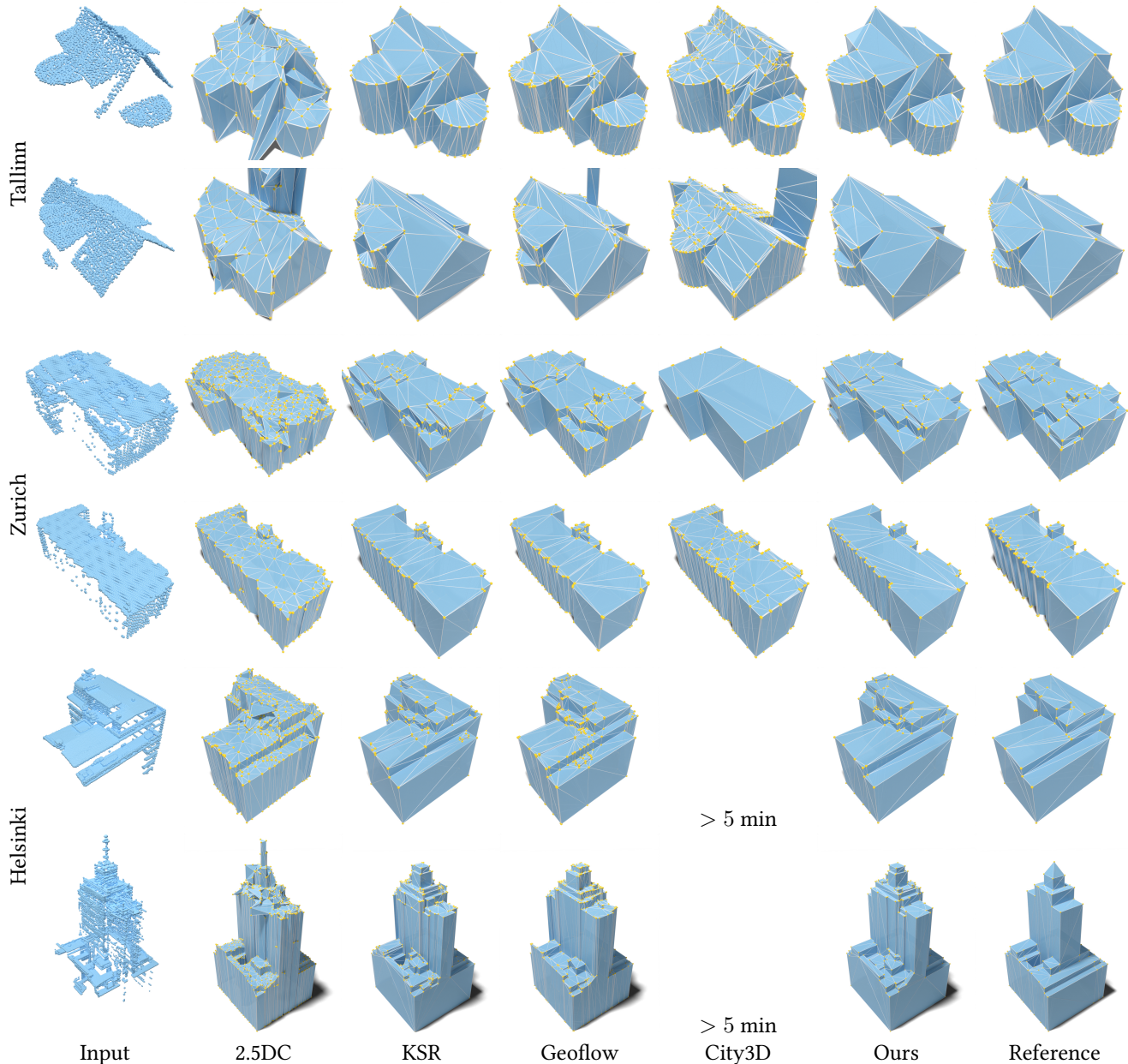


Figure 6. **Visual comparison.** The meshes produced by our algorithm capture the various roof components, even on complex buildings while being simple and regular. In contrast, meshes returned by prior methods are overly complex with respect to the reference models. KSR, Geoflow and City3D generate numerous tiny facets in between the roof sections while 2.5DC suffers from the presence of many visual artifacts. The two top buildings (respectively the third and fourth top buildings, and the two bottom buildings) originate from the *Building3D Tallinn dataset* (respectively the *Zurich dataset* and the *Helsinki dataset*).

4.3. Ablation

To validate our pipeline we ablate the main design choices and reconstruct a subset of the Tallinn building models. We show the results in Tab. 2. We first remove the discontinuity lines from the input of the 2D polygonal partition. This results in a simpler partition without the inner vertical discontinuities, and finally in models with slightly fewer

vertices and facets, but with an increased average distance from the input point cloud of around 4 cm. Conversely, disabling the two optimization procedures of Sections 3.3 and 3.4, or one of them only, leads to more complex models that satisfy less geometric guarantees but exhibit a slightly better accuracy, *i.e.* less than 1 cm on average on the two Chamfer distances.



Figure 7. **Textured reconstructions of Le Luc, France.** Both airborne Lidar point clouds and cadastral maps available online [13] were used as input to generate 3D building models, prior to the texturing process. Best viewed in electronic version.

Table 2. **Ablation study.** We ablate the main design choices of our pipeline (row 1) and evaluate reconstructions over a subset of 1,000 buildings of the Tallinn dataset. We remove vertical discontinuities from the construction of the 2D kinetic partition (row 2), both regularization steps (row 3), or apply only the vertical (row 4) or horizontal (row 5) regularization.

	Complexity			Accuracy	
	$ V $	$ F $	$E_{<50\text{ cm}}$ [%]	$CD_{\text{Inp} \rightarrow \text{Rec}}$ [cm]	$CD_{\text{Rec} \rightarrow \text{Ref}}$ [cm]
Ours	32.7	58.3	5.01	9.11	12.8
No vertical discontinuities	29.5	53.1	4.89	13.5	10.5
No regularization	38.7	66.7	8.13	8.13	12.6
Only vertical reg.	34.2	62.1	6.41	8.54	12.4
Only horizontal reg.	34.0	59.9	5.3	8.02	13.1

4.4. Application to city texturing

The ability of our algorithm to produce simple and regularized meshes is particularly interesting for simulation tasks, but not only. It is also valuable for visualization scenarios.

To this end, we adapt the inverse procedural modeling pipeline of Girard *et al.* [18]. Instead of modeling building geometries using remote imagery and shape grammars, we directly use 3D models generated by our pipeline as input of the texturation pipeline. Roofs are then textured based on their shape (flat, gabled, hipped, etc) using a library of predefined high-resolution textures, and ground-based imagery is used to extract the most similar element from a library of high-resolution facade textures, to preserve the architectural style of the building. Fig. 7 shows the result of such texturing process at a city scale.

4.5. Limitations

Our algorithm has a few limitations. Firstly, the 2.5D modelling approach yields LOD2.2 models and does not allow to reconstruct 3D building features such as roof overhangs and balconies (which would correspond to LOD2.3), that are yet present in many real-world cases. Besides, the robustness of our method relies upon the quality of the detected plane configurations. In particular, if the plane de-

tection technique misses one important planar element, then our reconstruction pipeline will be unable to reconstruct the related wall or roof section. This can make some near-vertical structures, such as steeples, hard to reconstruct, because they are represented by a limited number of points in airborne Lidar data. Curved structures, such as domes, are also approximated by a piecewise-planar geometry. Finally, our reconstruction pipeline assumes the input building footprints and Lidar scans align well. Misalignments problems are however frequent, especially when using online cadastral maps.

5. Conclusion

We present SimpliCity, a building reconstruction method that produces simple, regularized 3D models while offering a similar efficiency and fidelity to input data compared to prior pipelines. Our planimetric arrangement-based framework relies upon two key ideas: (i) local and global regularization of a 2D polygonal partition built from detected 3D planes, and (ii) extrusion of the partition with guaranteed planarity of roof sections and preservation of vertical discontinuities and horizontal rooftop edges.

In the future we would like to extend our optimization techniques to produce regularized LOD3 building models. Besides, we intend to investigate on the building segmentation and contouring problems, to correct issues induced by incorrect or incomplete building footprints. We also plan to release our reconstruction pipeline as a web service.

Acknowledgements

This research work was funded by the DGA GENESE project. The authors warmly thank Vincent Madelain for fruitful discussions, and Cédric Larrosa, Enora Fiker, Yacine Kacem for the proposed help for conducting experiments.

References

- [1] Bastian Albers, Martin Kada, and Andreas Wichmann. Automatic extraction and regularization of building outlines from airborne lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41, 2016. **3**
- [2] Dmitry Anisimov, Gennadii Sytov, Simon Giraudot, Jean-Philippe Bauchet, and Florent Lafarge. Shape regularization. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.6 edition, 2023. **3**
- [3] Jean-Philippe Bauchet and Florent Lafarge. Kippi: Kinetic polygonal partitioning of images. In *CVPR*, 2018. **3, 4**
- [4] Jean-Philippe Bauchet and Florent Lafarge. City reconstruction from airborne lidar: A computational geometry approach. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 2019. **2, 3**
- [5] Jean-Philippe Bauchet and Florent Lafarge. Kinetic shape reconstruction. *Trans. on Graphics*, 39(5), 2020. **1, 2, 6**
- [6] Filip Biljecki, Jantien Stoter, Hugo Ledoux, Sisi Zlatanova, and Arzu Çöltekin. Applications of 3d city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4), 2015. **1**
- [7] Alexandre Boulch and Renaud Marlet. Poco: Point convolution for surface reconstruction. In *CVPR*, 2022. **2**
- [8] Vasileios Bouzas, Hugo Ledoux, and Liangliang Nan. Structure-aware building mesh polygonization. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167, 2020. **2**
- [9] Anne-Laure Chauve, Patrick Labatut, and Jean-Philippe Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *CVPR*, 2010. **2**
- [10] Dong Chen, Ruisheng Wang, and Jiju Peethambaran. Topologically aware building rooftop reconstruction from airborne laser scanning point clouds. *Trans. on Geoscience and Remote Sensing*, 55(12), 2017. **2**
- [11] Jie Chen and Baoquan Chen. Architectural modeling from sparsely scanned range data. *International Journal of Computer Vision*, 78, 2008. **2**
- [12] Zhaiyu Chen, Hugo Ledoux, Seyran Khademi, and Liangliang Nan. Reconstructing compact building models from point clouds using deep implicit fields. *ISPRS Journal of Photogrammetry and Remote Sensing*, 194, 2022. **2**
- [13] Institut National de l'Information Géographique et Forestière. <https://geoservices.ign.fr>, 2024. **1, 8**
- [14] Li Deren, Yu Wenbo, and Shao Zhenfeng. Smart city based on digital twins. *Computational Urban Science*, 1, 2021. **1**
- [15] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. *Trans. on Information Theory*, 29(4), 1983. **3**
- [16] Michael Garland and Paul Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH*, 1997. **2**
- [17] Charles Gaydon. Myria3d: Deep learning for the semantic segmentation of aerial lidar point clouds, 2022. IGN (French Mapping Agency). **2**
- [18] Nicolas Girard, Cedric Larrosa, Willard Mapurisa, Frederic Trastour, and Yuliya Tarabalka. Brightearth city texturing: Faithful procedural 3d urban modeling from satellite and ground imagery. In *IGARSS*, 2023. **8**
- [19] GoogleEarth. <https://earth.google.com>, 2024. **1**
- [20] Gerhard Gröger and Lutz Plümer. Citygml–interoperable semantic 3d city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, 2012. **2**
- [21] Jiali Han, Lingjie Zhu, Xiang Gao, Zhanyi Hu, Liyang Zhou, Hongmin Liu, and Shuhan Shen. Urban scene lod vectorized modeling from photogrammetry meshes. *Trans. on Image Processing*, 30, 2021. **2**
- [22] Jin Huang, Jantien Stoter, Ravi Peters, and Liangliang Nan. City3d: Large-scale building reconstruction from airborne lidar point clouds. *Remote Sensing*, 14(9), 2022. **1, 2, 3, 6**
- [23] Jingen Jiang, Mingyang Zhao, Shiqing Xin, Yanchao Yang, Hanxiao Wang, Xiaohong Jia, and Dong-Ming Yan. Structure-aware surface reconstruction via primitive assembly. In *ICCV*, 2023. **2**
- [24] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *CVPR*, 2020. **2**
- [25] Martin Kada and Laurence McKinley. 3d building reconstruction from lidar based on a cell decomposition approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(3), 2009. **2**
- [26] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *Trans. on Graphics*, 32(3), 2013. **2**
- [27] Florent Lafarge and Clément Mallet. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. *International Journal of Computer Vision*, 99, 2012. **2, 5**
- [28] Florent Lafarge, Xavier Descombes, Josiane Zerubia, and Marc Pierrot-Deseilligny. Structural approach for building reconstruction from a single dsm. *Trans. on Pattern Analysis and Machine Intelligence*, 32(1), 2010. **2**
- [29] Li Li, Nan Song, Fei Sun, Xinyi Liu, Ruisheng Wang, Jian Yao, and Shaosheng Cao. Point2roof: End-to-end 3d building roof modeling from airborne lidar point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 193, 2022. **2**
- [30] Minglei Li and Liangliang Nan. Feature-preserving 3d mesh simplification for urban buildings. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173, 2021. **2**
- [31] Minglei Li, Peter Wonka, and Liangliang Nan. Manhattan-world urban reconstruction from point clouds. In *ECCV*, 2016. **2**
- [32] Minglei Li, Franz Rottensteiner, and Christian Heipke. Modelling of buildings from aerial lidar point clouds using tins and label maps. *ISPRS Journal of Photogrammetry and Remote Sensing*, 154, 2019. **2**
- [33] Yuan Li and Bo Wu. Relation-constrained 3d reconstruction of buildings in metropolitan areas from photogrammetric point clouds. *Remote Sensing*, 13(1), 2021. **3**
- [34] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. Globfit: Consistently fitting primitives by discovering global relations. *Trans. on Graphics*, 2011. **3**

- [35] Xinyi Liu, Yongjun Zhang, Xiao Ling, Yi Wan, Linyu Liu, and Qian Li. Topolap: topology recovery for building reconstruction by deducing the relationships between linear and planar primitives. *Remote Sensing*, 11(11), 2019. 2
- [36] Yujia Liu, Anton Obukhov, Jan Dirk Wegner, and Konrad Schindler. Point2building: Reconstructing buildings from airborne lidar point clouds, 2024. 3
- [37] Johann Lussange, Mulin Yu, Yuliya Tarabalka, and Florent Lafarge. 3d detection of roof sections from a single satellite image and application to lod2-building reconstruction. *arXiv preprint arXiv:2307.05409*, 2023. 2
- [38] Jisan Mahmud, True Price, Akash Bapat, and Jan-Michael Frahm. Boundary-aware 3d building reconstruction from a single overhead image. In *CVPR*, 2020. 2
- [39] Aron Monszpart, Nicolas Mellado, Gabriel J Brostow, and Niloy J Mitra. Rapter: rebuilding man-made scenes with regular arrangements of planes. *Trans. on Graphics*, 34(4), 2015. 3
- [40] Przemyslaw Musialski, Peter Wonka, Daniel Aliaga, Michael Wimmer, Luc Van Gool, and Werner Purgathofer. A survey of urban reconstruction. In *Computer Graphics Forum*, 2013. 1
- [41] Liangliang Nan and Peter Wonka. Polyfit: Polygonal surface reconstruction from point clouds. In *ICCV*, 2017. 2, 6
- [42] Gabriele Neyer. Line simplification with restricted orientations. In *Workshop on Algorithms and Data Structures*, 1999. 4
- [43] Gilles-Antoine Nys, Florent Poux, and Roland Billen. Cityjson building generation from airborne lidar 3d point clouds. *ISPRS International Journal of Geo-Information*, 9(9), 2020. 2
- [44] Sven Oesau, Florent Lafarge, and Pierre Alliez. Planar Shape Detection and Regularization in Tandem. *Computer Graphics Forum*, 35(1), 2016. 3
- [45] Sven Oesau, Yannick Verdie, Clément Jamin, Pierre Alliez, Florent Lafarge, Simon Giraudot, Thien Hoang, and Dmitry Anisimov. Point set shape detection. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.6 edition, 2023. 3
- [46] City of Helsinki. <https://kartta.hel.fi/>, 2021. 5, 6
- [47] Federal Office of Topography. <https://www.swisstopo.admin.ch/en/height-model-swissurface3d>, 2024. 1, 5
- [48] City of Zurich. <https://www.stadt-zuerich.ch/geodaten/download/3DDachmodellLoD2Jahresstand2015>, 2016. 5, 6
- [49] OpenStreetMap contributors. <https://www.openstreetmap.org>, 2017. 3
- [50] Jens Overby, Lars Bodum, Erik Kjems, and PM Iisoe. Automatic 3d building reconstruction from airborne laser scanning and cadastral data using hough transform. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(01), 2004. 2
- [51] Ravi Peters, Balázs Dukai, Stelios Vitalis, Jordi van Liempt, and Jantien Stoter. Automated 3D Reconstruction of LoD2 and LoD1 Models for All 10 Million Buildings of the Netherlands. *Photogrammetric Engineering and Remote Sensing*, 88(3), 2022. 1, 2, 6
- [52] Charalambos Poullis and Suya You. 3d reconstruction of urban areas. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2011. 2
- [53] Tahir Rabbani, Frank Van Den Heuvel, and George Vosselman. Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences*, 36(5), 2006. 2
- [54] David Salinas, Florent Lafarge, and Pierre Alliez. Structure-aware mesh decimation. In *Computer Graphics Forum*, 2015. 2
- [55] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer Graphics Forum*, 2007. 2
- [56] Raphael Sulzer, Loic Landrieu, Renaud Marlet, and Bruno Vallet. A Survey and Benchmark of Automatic Surface Reconstruction from Point Clouds. Preprint, 2023. 2
- [57] Carlos Vanegas, Daniel Aliaga, and Bedrich Benes. Building reconstruction using manhattan-world grammars. In *CVPR*, 2010. 2
- [58] Yannick Verdie, Florent Lafarge, and Pierre Alliez. LOD Generation for Urban Scenes. *ACM Trans. on Graphics*, 34(3), 2015. 2, 3
- [59] Vivek Verma, Rakesh Kumar, and Stephen Hsu. 3d building detection and modeling from aerial lidar data. In *CVPR*, 2006. 2
- [60] Julien Vuillamy, André Lieutier, Florent Lafarge, and Pierre Alliez. Simplification of 2D Polygonal Partitions via Point-line Projective Duality, and Application to Urban Reconstruction. *Computer Graphics Forum*, 41(6), 2022. 3
- [61] Biao Wang, Guoping Wu, Qiang Zhao, Yaozhu Li, Yiyuan Gao, and Jiangfeng She. A topology-preserving simplification method for 3d building models. *ISPRS International Journal of Geo-Information*, 10(6), 2021. 2
- [62] Ruisheng Wang, Jiju Peethambaran, and Dong Chen. Lidar point clouds to 3-d urban models: a review. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(2), 2018. 1
- [63] Ruisheng Wang, Shangfeng Huang, and Hongxin Yang. Building3d: An urban-scale dataset and benchmarks for learning roof structures from point clouds, 2023. 5, 6
- [64] Linfu Xie, Han Hu, Qing Zhu, Xiaoming Li, Shengjun Tang, You Li, Renzhong Guo, Yeting Zhang, and Weixi Wang. Combined rule-based and hypothesis-based method for building model reconstruction from photogrammetric point clouds. *Remote Sensing*, 13(6), 2021. 2
- [65] Biao Xiong, S Oude Elberink, and G Vosselman. A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. *ISPRS Journal of photogrammetry and remote sensing*, 93, 2014. 2
- [66] Biao Xiong, M Jancosek, S Oude Elberink, and G Vosselman. Flexible building primitives for 3d building modeling. *ISPRS Journal of Photogrammetry and Remote Sensing*, 101, 2015. 2
- [67] Siming Yan, Zhenpei Yang, Chongyang Ma, Haibin Huang, Etienne Vouga, and Qixing Huang. Hpnet: Deep primitive segmentation using hybrid representations. In *ICCV*, 2021. 2

- [68] Bisheng Yang, Wenxue Xu, and Zhen Dong. Automated extraction of building outlines from airborne laser scanning point clouds. *Geoscience and Remote Sensing Letters*, 10(6), 2013. [3](#)
- [69] Shengming Yang, Guorong Cai, Jing Du, Ping Chen, Jinhe Su, Yundong Wu, Zongyue Wang, and Jonathan Li. Connectivity-aware graph: A planar topology for 3d building surface reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 191, 2022. [2](#)
- [70] Mulin Yu and Florent Lafarge. Finding good configurations of planar primitives in unorganized point clouds. In *CVPR*, 2022. [2](#), [3](#)
- [71] Lukas Zebedin, Joachim Bauer, Konrad Karner, and Horst Bischof. Fusion of feature-and area-based information for urban buildings modeling from aerial imagery. In *ECCV*, 2008. [2](#)
- [72] Chaoquan Zhang and Hongchao Fan. An improved multi-task pointwise network for segmentation of building roofs in airborne laser scanning point clouds. *The Photogrammetric Record*, 37(179), 2022. [2](#)
- [73] Xiaowei Zhang and Daniel Aliaga. Procedural roof generation from a single satellite image. In *Computer Graphics Forum*, 2022. [2](#)
- [74] Qian-Yi Zhou and Ulrich Neumann. 2.5 d dual contouring: A robust approach to creating building models from aerial lidar point clouds. In *ECCV*, 2010. [1](#), [2](#), [5](#), [6](#)
- [75] Qian-Yi Zhou and Ulrich Neumann. 2.5d building modeling with topology control. In *CVPR*, 2011. [2](#)
- [76] Qian-Yi Zhou and Ulrich Neumann. 2.5d building modeling by discovering global regularities. In *CVPR*, 2012. [2](#), [3](#)
- [77] Lingjie Zhu, Shuhan Shen, Xiang Gao, and Zhanyi Hu. Large scale urban scene modeling from mvs meshes. In *ECCV*, 2018. [2](#)