Topological-based roof modeling from 3D point clouds

Dobrina Boltcheva[1], Justine Basselin[1,2], Clément Poull[1], Hervé Barthélemy[2], Dmitry Sokolov[1]
[1]Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France
[2]RhinoTerrain, Nancy, F-54000, France

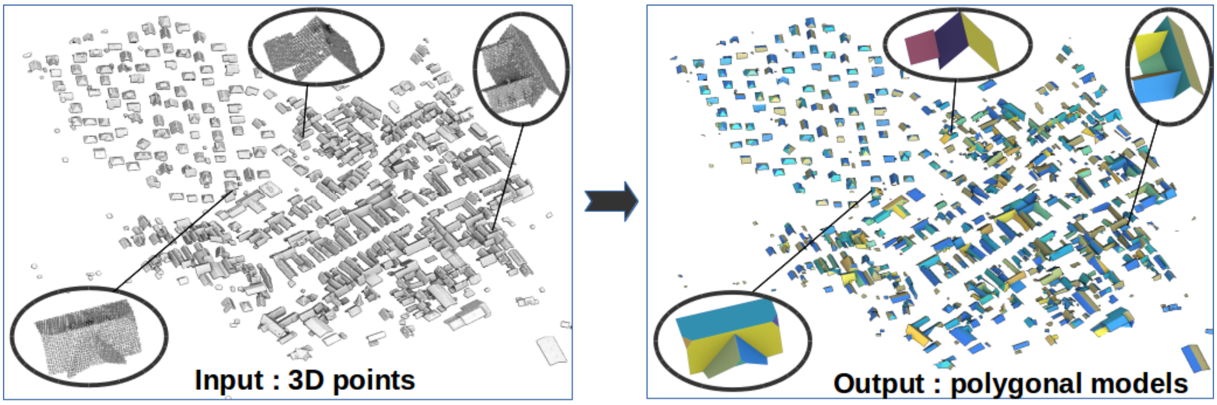

Figure 1: We address the problem of constructing polygonal 3D roof models from previously classified LIDAR point data.

Abstract

Automatic extraction of building roofs from remote sensing data is important for many applications including 3D city modeling, urban planning, disaster management, and simulations. In this paper, we propose an automatic workflow for roof reconstruction by polygonal models from classified high-density LIDAR data. Roof planes are initially delineated by a segmentation algorithm combining a robust Hough-based normal estimator and a region growing strategy. Then, each roof is modeled by a 2D α-shape mesh which is used to discover not only building outline but also all ridges defined by intersecting roof planes, without any geometrical calculations. The mesh directly encodes the topological relations between neighboring planes which allows us to build the final polygonal model straightforwardly. This topological approach makes our solution more simple and robust than existing methods which mostly extract the intersection lines by means of geometrical computations. Experimental results show that the proposed workflow offers a high success rate for extraction at plane level (94% completeness, 92.7% correctness, 90.8% quality) when LIDAR point density is sufficiently high.

Keywords
automatic roof modeling, feature extraction, topology graph, polygonal model, 3D point cloud

1 INTRODUCTION

The digitization of real objects is increasingly used in fields such as urban planning, architecture, disaster management and homeland security. Acquisition tools such as airborne light detection and ranging (LiDAR) scanners make it possible to produce digital representations of entire cities in the form of 3D point clouds sampling the surfaces of the objects in the environment.

Despite the high degree of maturity achieved by digitization techniques, effective computational solutions for preprocessing and reconstruction from these measurements are rare and ill-adapted to the complexity of the environment (complex building structures and entire cities).

Today, the process of creating a digital model from such data is long, tedious and essentially manual. In this reverse engineering process, the human operator manually draws 3D model elements as close as possible to the point cloud.

Although significant effort has been put into the development of automatic and semi-automatic methods, which are currently appearing on the market, no solution proposed so far meets all industrial requirements in terms of precision, accuracy and efficiency. This is because the reconstruction of 3D building models is a complex task that requires a workflow of several processing steps such as classification, outline extraction, segmentation, feature recognition, hypothesis generation and verification, geometric modeling and construction, adjustment and refinement.

In addition, the reconstructed models must respect a number of structural constraints (planarity of roof segments, horizontal roof ridges, symmetry, etc.), which cannot be retrospectively integrated into existing solutions.

Despite the acquired knowledge, there is still a significant number of unsolved problems coming from: gaps in the data (due to occlusions or unwanted reflections and absorption); noise and outliers; limited resolution and variable point density; high variability and complexity of building shapes in urban areas, to name a few.

In this work, we address the particular problem of constructing (creating) polygonal 3D roof models from previously classified LIDAR point data. Our method follows a pure data-driven approach and is fully automatic, except for the parameters which must be chosen with care by a human operator. The proposed method strives for building roof models with a level of detail (LOD2), as defined in the CityGML standard [18], e.g. detailed roof structures but without superstructures (such as chimneys, dormers, *etc.*). Moreover, each individual roof is assumed to be a set of flat planes.

A typical workflow of data-driven reconstruction methods consists of the following three steps:

1. Building points are aggregated to planar patches (segments) which represent roof facets.
2. The resulting segments are then combined to extract building modeling features or cues such as intersection and step lines, building outlines, corners, simple surface primitives, *etc.*.
3. Finally, 3D building models are constructed based on the extracted modeling features and subsequently regularized.

We are following this pipeline, improving steps 2 and 3. Our originality comes from the fact that we fix in priority the topological properties of the resulting model, which are known to be difficult to discover using geometrical approaches.

In addition, this combinatorial approach makes it easier to handle the step edges corresponding to height jumps.

2 RELATED WORK

Numerous building reconstruction approaches have been proposed in the last two decades and are presented in various surveying articles [37, 19, 8].

The model-driven approaches try to fit certain shapes to the data, while the data-driven approaches try to extract shapes present in the data. Although the model-driven methods are robust, their performance is limited to known models [46, 51, 50]. The data-driven methods work, in theory, for any rectilinear building shapes.

Here, we provide a brief overview of some data-driven approaches that are appropriate to our work; the sections correspond to the above pipeline.

2.1 Segmentation of planar patches

We can classify the (data-driven) building reconstruction methods into four categories w.r.t the approach do detect planar regions in point clouds:

Hough transform [20] is one of the earliest methods used for building reconstruction [27], and it has been enhanced and tuned many times [21, 32, 28].

RANSAC is another method that is applied frequently [16, 9]. It tends to be more robust than Hough transform. A comparison between these methods is presented in [45] for automatic segmentation of planar areas from point clouds.

Region growing gains its place thanks to its simplicity and efficiency. In contrast to Hough transform and RANSAC, it is a local segmentation technique. In [2] surface growing is applied to a regularized raster of building points to construct planar segments by a cell aggregation technique. Further surface growing based segmentation methods in the context of 3D building reconstruction are, for example, used in [4, 1, 44, 52].

Another viable (but costly) option is to perform a *global optimization*. In [26], flat roof segments are determined by minimizing an energy function formulated as a multiphase level set.

2.2 Model generation

After a set of segments has been determined, modeling cues can be extracted for the subsequent construction of 3D building models. For this, *intersection lines*, *step lines*, and *building outlines* are frequently extracted. Although building outlines can be considered as a special type of step lines, they are often determined in a separate processing step. There are several approaches that focus on generating a construction draft and simplifying or regularizing it. These approaches are, for example, based on RANSAC [23], α-shape [36, 5, 38], structured grids [52, 44], or line simplification such as Douglas-Peucker [12].

For the detection of step edges, height discontinuities between adjacent segments are searched in [48]. In [41], step edges are determined based on statistical tests and robust estimation. A so-called compass line filter (CLF) is proposed in [17]. It determines the local edge orientation for each step edge.

The ridge lines are usually directly obtained by the intersection of two planes that are derived from a pair of adjacent segments. Other modeling cues are, for example, extracted in [33]. Here, segments are enlarged to roof faces by intersecting all segments, regardless of their adjacency to each other. An approach to detect modeling cues in form of ridge lines utilizing RANSAC is presented in [15].

Most data-driven reconstruction methods generate a 3D polyhedral model directly based on the modeling cues, [41, 32, 36, 52, 49]. For this, the extracted lines are extended and connected with each other so that each roof surface is bounded by a closed sequence of connected line segments. Further faces with a vertical orientation are then added at step edges so that each edge in the polyhedral model becomes part of two surfaces defining line segment sequences. The implementation of such a direct polyhedral model generation method often comes, however, with quite a few problems because there are usually some ambiguities how the line segments are to be connected to guarantee the planarity of each polyhedral face without any gaps in between. There are methods using 2.5D triangulation [11], but it does not guarantee the correct topology of the model (step edges).

2.3 Model regularization

Since building models derived from data-driven reconstruction approaches resemble very closely the (imperfect) input data, several regularization and optimization methods have been developed. For this, most of the methods described above incorporate the main orientation of the building and support orthogonal and parallel structures.

An adjustment model that considers the building topology to improve the shape of a building model is, for example, proposed in [40]. In [52], global regularities are incorporated during the construction. This includes orientation and placement regularities between two roof surfaces, parallelism and orthogonality between building outlines and the normal of their owner planes, and regularities between two boundary edges in terms of their height and position. In order to automatically determine independent and consistent constraints, a greedy algorithm is proposed in [6] while Grobner bases are used in [29]. In [43] and [24], a refinement of building models is proposed based on aerial images from which building edges are detected and incorporated in the reconstruction process. An implicit regularization process in the framework of MDL in combination with hypothesize and test (HAT) is, for example, proposed in [22].

3 PROPOSED METHOD

Our method takes as input a set of clusters of 3D points where each cluster corresponds to the roof of an individual building or a set of adjacent buildings. It provides a 3D geometric and topological roof model where each planar patch is represented by a flat polygon and these polygons are consistently sewn together. These models are suitable for the subsequent reconstruction of polyhedral buildings by geometric extrusion or as an initialization for model-based fitting approaches.

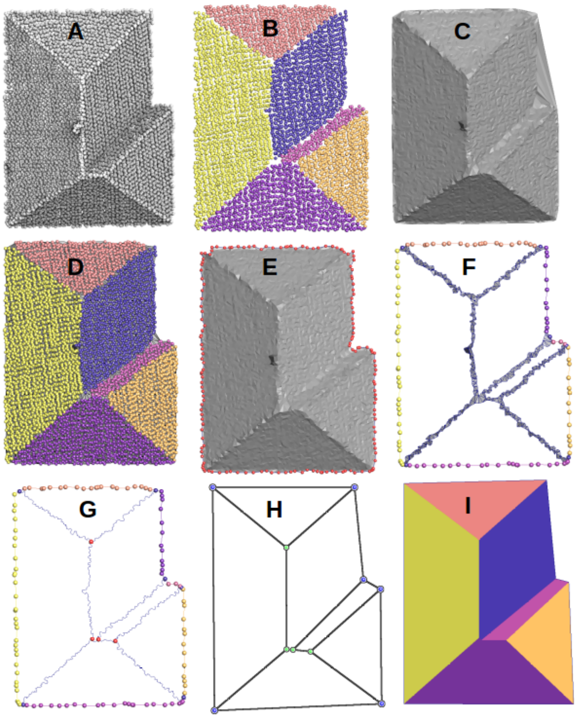

Figure 2: Detailed workflow. A: input points; B: delineated planar segments; C: Delaunay triangulation; D: α-shape; E: roof outline; F: ridge lines triangles; G: raw topological graph; H: simplified graph; I: polygonal mesh.

We follow the pipeline depicted in §1:

1. The planar roof segments are first extracted from each cluster using a robust Hough-based normal estimator and a region growing approach.

2. Then, each cluster of 3D points is projected to the xy-plan and its α-shape is built from its Delaunay triangulation in 2D. The initial building outline, in terms of edges, is extracted from the outmost triangles. We extract roof ridge lines and corners directly from the mesh using the set of triangles belonging to two or three segments. This approach allows us to build at first an accurate topological model (graph) where each vertex corresponds to the intersection of exactly three planar roof patches. Each edge corresponds to the intersection of adjacent planes and each facet corresponds to a roof plane.

3. Finally, we regularize the model by detecting the step edges and ensuring the polygons planarity.

The rest of the section is organized as follows: §3.1 describes the plane detection step, then §3.2 explains the model generation step, and, finally, §3.3 detailed the model regularization step. Figure 2 provides a visual representation of the method as a number of different entities computed during the processing (A to I).

3.1 Plane detection

Our method for plane detection processes each building cluster individually and takes into account the coordinates of the input points and the estimated surface normal vectors. We used a robust Hough-based algorithm provided by Boulch [7] which approximates the tangent plane at each point by considering its 30-neighborhoods. This method is particularly well suited to our data because it does not smooth the normal vectors at the intersection lines between adjacent planes.

Then, a *Region Growing* algorithm, provided by the Point Cloud Library [35], is used to merge the points that are close enough in terms of a *smoothness constraint* defined as the deviation between normals of the points. In order to compute the Cartesian equation of each detected planer patch, we compute the plane normal vector as the average of the normals of its points.

At the end of this step, each building cluster has its points labeled with a value corresponding to the planar patch to which it belongs, as shown in Fig.2-B. In addition, we keep the computed plane equations, as a text file, which will be used in a next processing step.

3.2 Model generation

Based on the planar patch segmentation performed in the previous step, this step deals with the construction of a roof *shape descriptor* in the form of a graph consisting of the *roof's outer edge*, the *ridge lines* and their mutual intersections, referred to as *corners*. It essentially involves a surface mesh generation algorithm allowing to build a triangular mesh connecting all the input points which can then be used to extract the coarse roof model. This initial model will be geometrically optimized during the last step, in order to deliver a valid polygonal 3D roof model.

First, we project the input 3D points to the xy-plane by ignoring the z coordinate. Then, we build the *2D Delaunay Triangulation* of the points which is a mesh of their *convex hull* and can be embedded in 3D, as shown in Fig.2-C.

Since the building footprints are rarely convex polygons, this initial mesh has to be *carved* from outside in order to remove the triangles spanning the concave area. This brings us exactly to the definition of the α-shape of the points which is usually computed from the Delaunay mesh by removing the triangles having an edge longer than the parameter $\alpha > 0$, [14]. Indirectly we are performing a Delaunay sculpting approach as introduced in [13] and [30].

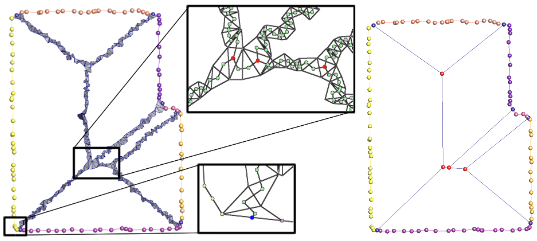

Figure 3: On the left, dual ridge edges. The *corner* vertices are in red, the *connection* vertex is in blue. On the right, simplified ridges by removing *ordinary* vertices.

In practice, we have implemented an erosion and boundary discovery algorithm. It stats by constructing the convex roof boundary as the list of outmost edges (*i.e.* incident to only one triangle in the Delaunay mesh). Then, it considers the shortest edge and checks if it is smaller than the threshold value α. If yes, the edge is kept and the algorithm continues with one of its adjacent edges. Otherwise, the edge is removed from the boundary list and is replaced by the two other edges of the current triangle. The algorithm stops when all the edges of the boundary list have been checked.

The most critical aspect of this algorithm consists in the selection of an optimal value for the parameter α, since it depends directly on point density and the desired level of detail in the boundary extraction. Although there exist a recent attempt [38] to incorporate the density variation information in the algorithm in the particular context of building modeling.

At this stage of the processing, we have a dense triangular mesh which perfectly fits to the input data, as depicted in Fig.2-D. Note that we keep the labels at the vertices computed at the previous stage. They are shown in different colors and indicate to which plane each vertex belongs to. This information will be used to discover the ridge lines and the corners between planes of the polygonal model.

In addition, the mesh edges of the concave border are also tagged with a specific value, their vertices are shown in red in Fig.2-E. We assume that the building footprint (outline) has only one connected component. Otherwise, processing stops and the input building cluster must be refined in order to correctly separate the various connected components.

In deferential geometry, the ridge lines are defined as the locations of local extrema of the principal curvature of the surface, whereas their intersecting *corners* are locations where the principal curvature vanishes.

In our particular case, the ridge lines are defined by the intersection of two adjacent planar patches, while the corners corresponds to the intersections of exactly three of them. These topological features are easily accessible from the mesh as the set of triangles having more than two different vertex labels, as shown in Fig.2-F.

This simple observation, allows us to design a robust combinatorial algorithm in 2D which essentially consists in a simple mesh traversal and does not use any user-given parameters.

This is in sharp contrast with most of the concurrent methods which usually relay on 3D geometrical computations (adjacent planes searching, planes intersection infinite line computation, ridge line end points searching, *etc.*) which are known to be tricky to be robustly implement with floating point numbers [10].

In addition, by choosing to work as long as possible in 2D, we avoid many complex 3D configurations and defer the consideration of height jump edges to the last step of the workflow.

In practice, there is no technical difficulty to build the topological graph from the mesh as far as the mesh is encoded in an appropriate data structure, for example the half-edge data structure [31, 25].

The algorithm starts by adding to the graph the vertices and the edges of the outline border which have been previously tagged in the mesh. Then it adds the *dual edges* of the ridge triangles that have been previously tagged (spanning two or three planes). This consists in adding a vertex in the center of each triangle and connecting the barycenters of the triangles that share a common edge (see zoom in Fig.3). The algorithm also records, for each graph vertex, an attribute composed of the labels of the involved planes. In 2D, there are three possible configurations, as follows: If a vertex has exactly three labels, it is tagged as a *corner* and will be locked up and kept forever in the graph. If a vertex has exactly two labels, it is an *ordinary* vertex and could be removed during the optimization step. If a vertex has a unique label, it is a *connection* vertex and must be preserved. It is connected to the outline border by projecting it to the closest boundary edge (which is necessarily one of the edges of the triangle it belongs to) (see zoom in Fig.3).

The resulting raw topological graph is shown in Fig.2-G where the *corner* vertices are depicted in red, while the *connection* ones are in blue. The *ordinary* vertices are not highlighted for better visibility but the edges between them are drawn in blue.

Notice that, at this stage of the processing, the topological graph is composed of many small edges which have been extracted from the dense α-shape mesh. However, we are able to anticipate the model optimization by simply removing all *ordinary* vertices, which allows is to already simplify the internal ridge lines, as shown in Fig.3.

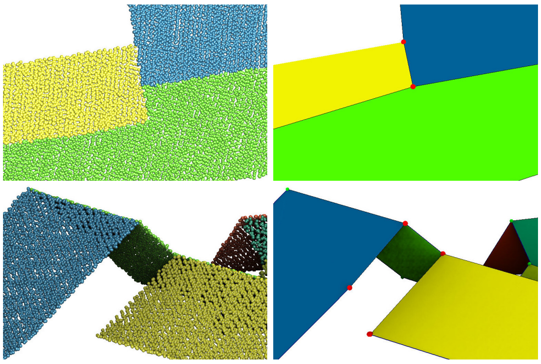

Figure 4: Model regularization step. **Top row:** input points as seen from above and corresponding 2D polygons after the simplification. **Bottom row:** side view of the original input points and the regularized model. Red dots show the vertices that were duplicated.

3.3 Model regularization

The model computed by the previous steps already has the correct topology but needs to be geometrically optimized to become an acceptable roof model. First we need to simplify the polygon boundaries, then we have to deal with the planarity constraint as well as the detection of height skip lines.

There are several methods in the literature to simplify polylines. While Douglas-Peucker's algorithm [12] is the best known, we have implemented Visvalingam's algorithm [47], because it is more efficient. The idea is very simple: we gradually remove points with the least-perceptible change, as in [34].

We have a polygonal (non-planar) mesh and the Cartesian plane equations. Each vertex of our model to be regularized is incident to one, two or three facets. According to the connectivity we move the vertices:

- *One incident facet:* we project it onto the corresponding plane.
- *Two incident facets:* we project it onto the intersection line.
- *Three incident facets:* we move it into the intersection of the corresponding planes.

While the projection is conceptually simple, care must be taken. Refer to the bottom left image of the Fig. 4 for an illustration. Yellow and blue planes do not intersect each other, yet the 2D α-shape is not aware of that.

Consider the vertices highlighted in red in top right image of the figure; if we project them onto both planes simultaneously, we get an inconsistent result. Fortunately, it is very easy to detect this inconsistency.

Gross density	Average density	Beam divergence	Accuracy x, y	Accuracy z	Mean distance between points
28 pts/sqm	30 pts/sqm	0,25 mrad	10 cm	6 cm	20 cm

Table 1: Lidar parameters.

Indeed, an inconsistency can be detected thanks to the distance between the original position of the point and its possible new position. If this distance exceeds a certain threshold then it's an inconsistency. If the inconsistent point has a valence of three, then among the adjacent planes, we identify a pair of planes that appear to be the most parallel (the ones that will generate the greatest inconsistency). And we duplicate the point so as to separate the two planes. If it has a valence of two then we simply duplicate the point and separate the two planes as shown in the bottom right image. So, the regularization step can be seen as the following loop:

1. project all vertices onto the corresponding planes;
2. if there are no inconsistent configurations, stop;
3. else duplicate the inconsistent vertices and go to the projection step.

The remarkable aspect here is the fact that our algorithm detects and recovers the inconsistencies that correspond to the step edges (height jump lines). The step edge detection is a hard problem and our combinatorial approach is more robust than the geometry-based algorithms like 3D α-shape or snapping outlines of individual roof panes. Fig. 5 presents some examples of the regularization algorithm.

4 EXPERIMENTAL RESULTS

Data set description

The data set used in our tests was captured over Breuschwickersheim of the Eurometropolis of Strasbourg in France using the LMS Q780 Riegl laser scanner (see Tab. 1). It covers an area of approximately $1000\,m \times 1000\,m$.

The classification of the point cloud is obtained during the acquisition by return wave analysis. From this classification, we were able to extract a set of 2.5M points corresponding to the building roofs of the entire area (Fig.1). This single point cloud has been further delineated into clusters corresponding to individual buildings. We obtained 445 clusters using the *Euclidean Cluster Extraction* algorithm provided by the Point Cloud Library [35]. Any other clustering algorithm could be used for this purpose. Each resulting cluster corresponds to the roof of a single building or a set of adjacent buildings (Fig.1). Note that, in very densely built-up areas, the latter configuration may be the norm and no automatic tool can determine a limit between neighboring buildings if there is no gap between them, without cadastral information. Since we are not interested in the problem of building detection, we assume that the building delineation is correct. We have kept the large clusters as they were delivered, without cutting them manually. This allows us to provide reasonably challenging inputs to our roof modeling method, presented here.

As a reference data, a human operator has built manually the polygonal models for every cluster shown in Fig.1 using the RhinoCapture software [39]. An example of a reference roof is given in Fig. 6.

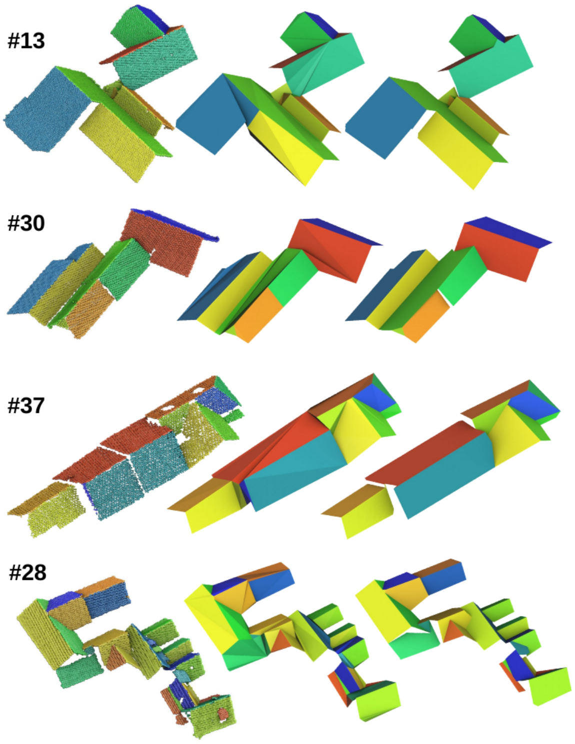

Figure 5: Models with step edges. First column: annotated points; Second column: topological graph; Third column: model after regularization.

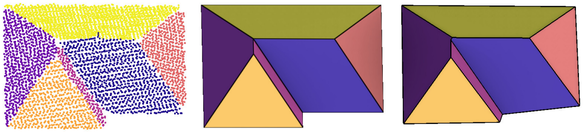

Figure 6: **Left:** labeled input point set; **Middle:** manually constructed roof model; **Right:** automatic reconstruction.

	C_m	C_r	Q_l
Mean	94.0	92.7	90.8
Median	100	100	100
First quartile	91.2	89.6	80.8

Table 2: Accuracy assessment of the plane detection over the entire data (445 building clusters).

Quantitative evaluation

We have used the evaluation system proposed in [42, 3] which assumes that the roofs are represented by polygonal models and each individual roof consists of a set of flat planes. We have carried out an *object-based evaluation at plane level* since, let us recall, our method addresses only the problem of roof modeling by detecting planes in already segmented building clusters.

In object-based evaluation, *completeness* (C_m), *correctness* (C_r), and *quality* (Q_l) are estimated by counting the number of true positive (TP), false positive (FP) and false negative (FN) planes in the extracted results, as follows : $C_m = TP/(TP+FN)$, $C_r = TP/(TP+FP)$ and $Q_l = TP/(TP+FN+FP)$. Ideally, these values should be maximum at 100%. The *completeness* indicates the detection rate and is the percentage of entities in the reference data that were detected. The *correctness* indicates how well the detected entities match the reference data and is closely linked to the false alarm rate. The *quality* provides a compound performance metric that balances *completeness* and *correctness*.

As depicted on Table 2, the average completeness, correctness and quality metrics, computed on the entire dataset (445 clusters of buildings), were respectively 94%, 92.7% and quality 90.8%.

In addition to these quantitative results, qualitative analysis is also presented via visualization (see Fig.7).

Table 3 gives detailed evaluation metrics for the clusters depicted in Fig.5, Fig.7 and Fig.8. All of the clusters corresponding to individual buildings were perfectly reconstructed (see Fig.7). Large clusters corresponding to adjacent buildings were also correctly handled (see Fig.7:#10,#23). In the case of clusters presenting step edges (Fig.5), some true planes were missed and some roof planes were wrongly constructed. The last four clusters (Fig.8) are considered as fail cases which are mainly due to poor plane segmentation.

Discussion

The proposed method strives for building roof models with a level of detail LOD2, as defined in the CityGML standard [18]. This means that we look for a detailed roof structures without superstructures (e.g., chimneys, dormers, etc.).

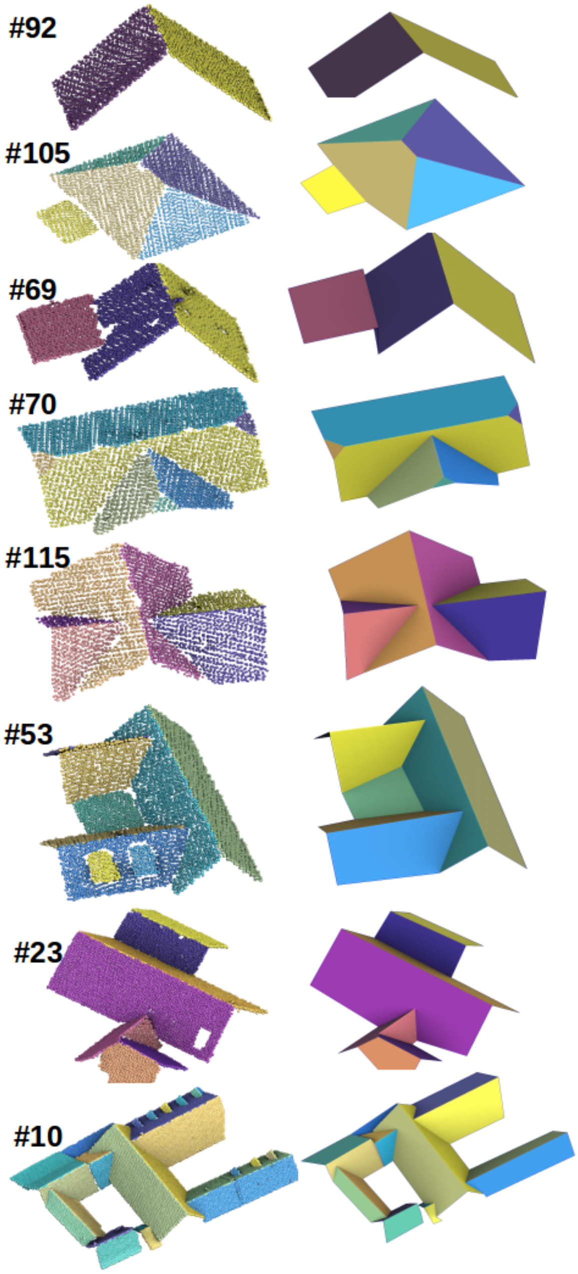

Figure 7: Resulting models for 8 clusters.

The small roof superstructures such as chimneys, antennas and various noisy components are mostly removed by the *plane segmentation algorithm* (step 1).

However, some small roof extensions and detail roof structure were correctly modeled as shown in Fig.7:#105.

The dormers are roof elements defining small planes which are topologically *inside* the primary roof planes. Large dormers are recognized and accurately delineated by our step 1 algorithm, as shown on Fig.7:#53. But these structures are usually removed by the *model generation algorithm* (step 2).

#	#ref	TP	FP	FN	C_m	C_r	Q_l
92	2	2	0	0	100	100	100
105	5	5	0	0	100	100	100
69	3	3	0	0	100	100	100
70	7	7	0	0	100	100	100
115	6	6	0	0	100	100	100
53	7	7	0	0	100	100	100
23	8	8	0	0	100	100	100
10	18	18	0	0	100	100	100
13	13	13	1	0	100	92.9	92.9
30	9	8	0	1	88.8	100	88.8
37	13	11	1	2	84.6	91.7	78.6
28	19	18	1	1	94.7	94.7	90
151	7	7	4	0	100	63.6	63.6
214	3	3	2	0	100	60	60
215	9	8	0	1	88.9	100	88.9
256	2	2	1	0	100	66.7	66.7

Table 3: Detailed evaluation metrics for the roofs in Fig.7, Fig.5 and Fig.8.

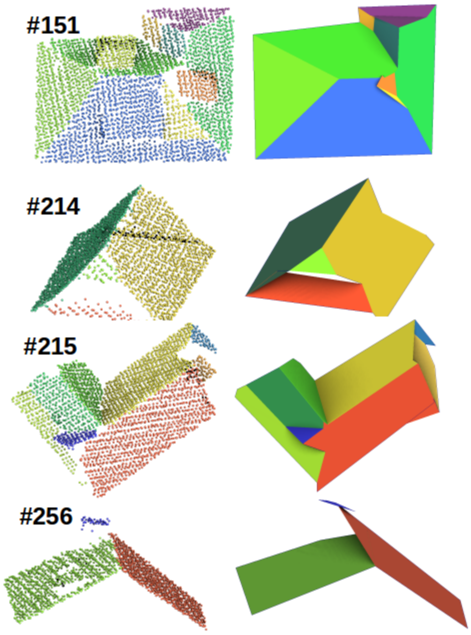

Figure 8: Examples of failure cases due to incorrect plane segmentation.

Let us recall that our *model generation algorithm* works in 2D, which allows us, on one hand, to design a robust combinatorial algorithm avoiding many 3D problems, on the other hand, to postpone the 3D embedding problems due to height jumps. This makes our method able to handle roofs with simple geometry quite efficiently. For simple flat, gamble and hip roofs, the algorithm achieved 100% completeness, correctness and quality. Our data contains 40% of this kind of structure. For cross-fipped, cross-gabled, hip and valley and intersecting roofs, the method achieved 98% of perfect models. These shapes constitute about 30% of our dataset (Fig.7). As show on Fig.5, most of the complex structures with height jumps are correctly optimized by our *model regularization* algorithm. The remaining 30% are complex structures, 20% of them are successfully handled and the resulting models are acceptable for further use (see Fig.7:#10,#23), while 10% of our data are real failures which are mostly due to poor plane segmentation, as shown in Fig.8.

5 CONCLUSION

In this paper we propose a new workflow for the reconstruction of building roofs from pre-classified LIDAR point data. The method takes as input a cluster of 3D points corresponding to the roof of an individual building. It delivers a 3D polygonal model where each planar roof patch is represented as a flat polygon and their ridge and corner intersections are topologically consistent. The main contribution is a combinatorial approach that is more robust than the geometrical methods trying to snap intersection lines. Having the combinatorial information, we easily reconstruct step edges during the regularization phase.

Experimental testing has shown that the proposed pipeline successfully deals with challenging input data. It achieves a plane detection rate of 94% and a quality rate of 90%.

We believe that the resulting models will be suitable for further polyhedral building reconstruction through geometric extrusions, for example. Moreover, they could also be useful for the initialization of model-driven methods for roof modeling.

6 ACKNOWLEDGMENTS

Our thanks to the company RhinoTerrain, for providing us with several datasets and for data modelling.

7 REFERENCES

[1] ABDULLAH, S., AWRANGJEB, M., AND LU, G. Lidar segmentation using suitable seed points for 3d building extraction. *Int. Arch. of the Phot., R. Sensing & S. I. Sciences* (2014).

[2] ALHARTHY, A., AND BETHEL, J. Detailed building reconstruction from airborne laser data using a moving surface method. *Inter. Archives of Photogrammetry and Remote Sensing 35* (01 2004).

[3] AWRANGJEB, M., AND FRASER, C. An automatic and threshold-free performance evaluation system for building extraction techniques from airborne lidar data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 7*, 10 (2014), 4184–4198.

[4] Awrangjeb, M., and Fraser, C. Automatic segmentation of raw lidar data for extraction of building roofs. *Remote Sensing 6* (05 2014), 3716–3751.

[5] B. Albers, M. Kada, A. W. Automatic extraction and regularization of building outlines from airborne lidar point clouds. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 41* (2016).

[6] Boge, M., Meidow, J., and Bulatov, D. Extraction and refinement of building faces in 3d point clouds. *The Inter. Society for Optical Engineering 8892* (10 2013), 88920V.

[7] Boulch, A., and Marlet, R. Fast and robust normal estimation for point clouds with sharp features. *Computer Graphics Forum 31* (08 2012), 1765–1774.

[8] Brenner, C. Building reconstruction from images and laser scanning. *International Journal of Applied Earth Observation and Geoinformation 6*, 3 (2005), 187 – 198.

[9] D. Chen, L. Zhang, J. L., and Liu, R. Urban building roof segmentation from airborne lidar point clouds. *Inter. Journal of Remote Sensing 33* (10 2012), 6497–6515.

[10] Devillers, O., and Guigue, P. Finite Precision Elementary Geometric Constructions. Tech. Rep. RR-4559, INRIA, Sept. 2002.

[11] Dong, C., Wang, R., and Peethambaran, J. Topologically aware building rooftop reconstruction from airborne laser scanning point clouds. *IEEE Transactions on Geoscience and Remote Sensing PP* (08 2017).

[12] Douglas, D., and Peucker, T. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Inter. J. for Geographic Information and Geovisualization 10*, 2 (1973), 112–222.

[13] Duckham, M., Kulik, L., Worboys, M., and Galton, A. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition 41* (10 2008), 3224–3236.

[14] Edelsbrunner, H. Alpha shapes - a survey. *Tessellations in the Sciences* (01 2010).

[15] Fan, H., Yao, W., and Fu, Q. Segmentation of sloped roofs from airborne lidar point clouds using ridge-based hierarchical decomposition. *Remote Sensing 6* (04 2014), 3284–3301.

[16] G. Forlani, C. Nardinocchi, M. S., and Zingaretti, P. Complete classification of raw lidar data and 3d reconstruction of buildings. *Pattern analysis and appli. 8*, 4 (2006), 357–374.

[17] G. Sohn, X. H., and Tao, V. Using a binary space partitioning tree for reconstruction polyhedral building models from airbome lidar data. *Photogrammetric Engineering and Remote Sensing 74* (08 2008), 1425–1440.

[18] Gröger, G., and Plümer, L. Citygml–interoperable semantic 3d city models. *ISPRS Journal of Photogrammetry and Remote Sensing 71* (2012), 12–33.

[19] Haala, N., and Kada, M. An update on automatic 3d building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing 65*, 6 (2010), 570 – 580.

[20] Hough, P. Method and means for recognizing complex patterns. *US patent 3*, 6 (1962).

[21] Huang, H., and Brenner, C. Rule-based roof plane detection and segmentation from laser point clouds. In *Joint Urban Remote Sensing Event* (2011), IEEE, pp. 293–296.

[22] Jaewook, J., Yoonseok, J., and Gunho, S. Implicit regularization for reconstructing 3d building rooftop models using airborne lidar data. *Sensors 17* (03 2017).

[23] Jarzabek-Rychard, M. Reconstruction of building outlines in dense urban areas based on lidar data and address points. *Inter. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 39* (07 2012), 121–126.

[24] Jarzabek-Rychard, M., and Maas, H.-G. Geometric refinement of als-data derived building models using monoscopic aerial images. *Remote Sensing 9* (03 2017), 282.

[25] Kettner, L. Using generic programming for designing a data structure for polyhedral surfaces. *Computational Geometry 13*, 1 (1999), 65–90.

[26] Kim, K., and Shan, J. Building roof modeling from airborne laser scanning data based on level set approach. *ISPRS Journal of Photogrammetry and Remote Sensing 66*, 4 (2011), 484–497.

[27] Maas, H., and Vosselman, G. Two algorithms for extracting building models from raw laser altimetry data. *J. of Photogrammetry and Remote Sensing 54*, 2 (1999), 153 – 163.

[28] Maltezos, E., and Ioannidis, C. Automatic extraction of building roof planes from airborne lidar data applying an extended 3d randomized hough transform. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences III-3* (06 2016), 209–216.

[29] Meidow, J., and Hammer, H. Algebraic reasoning for the enhancement of data-driven building reconstructions. *J. of Photogrammetry and Remote Sensing 114* (04 2016), 179–190.

[30] METHIRUMANGALATH, S., PARAKKAT, A., AND MUTHUGANAPATHY, R. A unified approach towards reconstruction of a planar point set. *Computers & Graphics 51* (05 2015).

[31] MULLER, D., AND PREPARATA, F. Finding the intersection of two convex polyhedra. *Theoretical Computer Science 7*, 2 (1978), 217 – 236.

[32] NOVACHEVA, A. Building roof reconstruction from lidar data and aerial images through plane extraction and colour edge detection. *Inter. Society for Photogrammetry and Remote Sensing* (01 2008), 51–56.

[33] OVERBY, J., BODUM, L., KJEMS, E., AND IISOE, P. Automatic 3d building reconstruction from airborne laser scanning and cadastral data using hough transform. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXIV* (01 2004).

[34] PARAKKAT, A., BONDI, U., AND MUTHUGANAPATHY, R. A delaunay triangulation based approach for cleaning rough sketches. *Computers & Graphics 74* (05 2018).

[35] PCL. Pcl. pointclouds.org.

[36] PETER, D., AND PFEIFER, N. A comprehensive automated 3d approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. *Sensors 8* (11 2008).

[37] R. WANG, J. P., AND CHEN, D. Lidar point clouds to 3-d urban models : a review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 11*, 2 (2018), 606–627.

[38] R.C. DOS SANTOS, M. G., AND CARRILHO, A. Extraction of building roof boundaries from lidar data using an adaptive alpha-shape algorithm. *IEEE Geoscience and Remote Sensing Letters 16*, 8 (2019), 1289–1293.

[39] RHINOTERRAIN. Rhinocapture. www.rhinoterrain.com.

[40] ROTTENSTEINER, F. Consistent estimation of building parameters considering geometric regularities by soft constraints. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 34* (01 2006).

[41] ROTTENSTEINER, F., TRINDER, J., CLODE, S., AND KUBIK, K. Automated delineation of roof planes from lidar data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36* (05 2012).

[42] RUTZINGER, M., ROTTENSTEINER, F., AND PFEIFER, N. A comparison of evaluation techniques for building extraction from airborne laser scanning. *J. of Selected Topics in Applied Earth Observations and Remote Sensing 2*, 1 (2009), 11–20.

[43] SOHN, G., JUNG, J., JWA, Y., AND ARMENAKIS, C. Sequential modelling of building rooftops by integrating airborne lidar data and optical imagery: preliminary results. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-3/W1* (05 2013).

[44] SUN, S., AND SALVAGGIO, C. Aerial 3d building detection and modeling from airborne lidar point clouds. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of 6* (06 2013), 1440–1449.

[45] TARSHA-KURDI, F., LANDES, T., AND GRUSSENMEYER, P. Extended ransac algorithm for automatic detection of building roof planes from lidar data. *The Photogrammetric Journal of Finland 21* (01 2008), 97–109.

[46] V. VERMA, R. K., AND HSU, S. 3d building detection and modeling from aerial lidar data. *Computer Society Conference on Computer Vision and Pattern Recognition 2* (02 2006), 2213 – 2220.

[47] VISVALINGAM, M., AND WHYATT, J. *Line generalisation by repeated elimination of the smallest area.* Cartographic Information Systems Research Group, 1992.

[48] VOSSELMAN, G. Building reconstruction using planar faces in very high density height data. *Int. Arch. Photogramm. Remot. Sens 32* (12 2000).

[49] XIAO, Y., WANG, C., LI, J., ZHANG, W., XI, X., WANG, C., AND DONG, P. Building segmentation and modeling from airborne lidar data. *International Journal of Digital Earth 8*, 9 (2015), 694–709.

[50] XIONG, B., ELBERINK], S. O., AND VOSSELMAN, G. A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing 93* (2014), 227 – 242.

[51] XU, B., JIANG, W., AND LI, L. Hrtt: A hierarchical roof topology structure for robust building roof reconstruction from point clouds. *Remote Sensing 9* (04 2017), 354.

[52] ZHOU, Q.-Y., AND NEUMANN, U. 2.5d building modeling by discovering global regularities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (June 2012), pp. 326–333.