PIECEWISE-PLANAR APPROXIMATION OF LARGE 3D DATA AS GRAPH-STRUCTURED OPTIMIZATION

S. Guinard[1], L. Landrieu[2], L. Caraffa[1], B. Vallet[1]

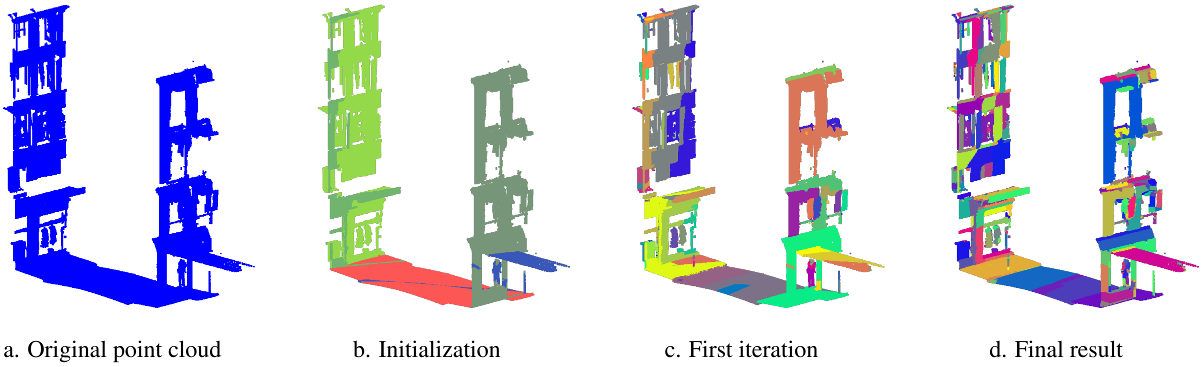
Univ. Paris-Est, IGN-ENSG, [1] LaSTIG-ACTE, [2] LaSTIG-STRUDEL, Saint-Mandé, France
(stephane.guinard, loic.landrieu, laurent.caraffa, bruno.vallet)@ign.fr

Commission II, WG II/3

KEY WORDS: point clouds, mesh, segmentation, piecewise-planar approximation, graph-cut

ABSTRACT:

We introduce a new method for the piecewise-planar approximation of 3D data, including point clouds and meshes. Our method is designed to operate on large datasets (e.g. millions of vertices) containing planar structures, which are very frequent in anthropic scenes. Our approach is also adaptive to the local geometric complexity of the input data. Our main contribution is the formulation of the piecewise-planar approximation problem as a non-convex optimization problem. In turn, this problem can be efficiently solved with a graph-structured working set approach. We compare our results with a state-of-the-art region-growing-based segmentation method and show a significant improvement both in terms of approximation error and computation efficiency.



a. Original point cloud b. Initialization c. First iteration d. Final result

Figure 1. Illustration of our method on a LiDAR scan of an urban scene. Our algorithm takes an input point cloud (a), and start by extracting planes using a RANSAC approach (b). These planes are used to initialize a graph-structured working-set algorithm. We represent the evolution of the approximation, after the first (c) and the last (d) iteration. Each color represents a different region.

1. INTRODUCTION

With the advance of modern acquisition technologies, 3D scans tend to be more and more dense thus contain more and more points. This allowed for an increase in the level of detail at the cost of an increased computational and memory load. Therefore, generalizing or simplifying 3D data would enable more efficient processing, interpretation and visualization of the data. The structure of anthropic objects, such as roads, roofs and facades, can be exploited to achieve this goal. Hence, LiDAR scans of anthropic objects would thus highly benefit from piecewise-planar approximation.

Most of the work in the literature focuses on the extraction of geometric primitives using a RANSAC-based approach (Schnabel et al., 2007) or a region growing clustering algorithm (Cohen-Steiner et al., 2004). These methods are then traditionally used as a baseline for 3D reconstruction (Peternell and Steiner, 2004; Chauve et al., 2010) or point cloud approximation (Fang et al., 2018).

Our idea differs from the literature, as we propose a formulation of the piecewise-planar approximation problem as a single non-convex optimization problem. Our method aims at producing a segmentation of point clouds into planes which is adaptive to the local geometric complexity. We also argue that a mesh structure is not necessary to perform a segmentation of 3D data, since we can represent a point cloud as a graph, structured with point adjacency (k-nearest neighbors for instance); and point adjacency does not require mesh information (even if mesh information can be used to build point adjacency).

2. STATE OF THE ART

Traditional approaches for approximation of 3D data consists in clustering points belonging to the same primitive, and then fitting models on each group. In this section, we present an overview of the state-of-the-art for these two problems.

2.1 Plane Fitting

The least squares method is the standard approach for fitting parametric model on 3D data. It consists in selecting the models for which the sum of square distance between the points and the model is minimal. This method has been used extensively

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W5-365-2019 | © Authors 2019. CC BY 4.0 License.

to find relevant primitives to approximate 3D point clouds as set of planes (Xie et al., 2003; Dzitsiuk et al., 2017). Tatavarti et al. (2017) partition their data into cubical regions and use a least square algorithm to fit a plane in each cell. This approach has also been used to reconstruct facades from 3D point clouds (Martinovic et al., 2015).

The RANSAC algorithm (Fischler and Bolles, 1981; Schnabel et al., 2007) is one of the most popular method for extracting planes in 3D point clouds. Given a dataset composed of inliers and outliers, this method works by iteratively fitting a mathematical model to a random subset of the data and checking whether the whole dataset is consistent with this mathematical model. In the case of piecewise-planar approximation of 3D data, the mathematical model will be a plane equation. This approach has been thoroughly investigated in the literature (Asvadi et al., 2016; Dzitsiuk et al., 2017). Holzmann et al. (2017) used a RANSAC-based approach for plane extraction in point clouds, and a reconstruction algorithm based on extracted planes to build a Manhattan-like structure. Moreover, Xu et al. (2015) introduced a weighted RANSAC approach with a soft threshold voting function, based on two weighted functions. It allows them to lower the number of spurious planes and improve segmentation quality.

Another way of finding planes in 3D data is to use a planarity geometric feature as detailed in Demantke et al. (2011). These features are based on the eigenvalues of the co-variance matrix of points' neighbors coordinates. Chauve et al. (2010) use a planarity feature to detect planes. Boulch et al. (2014) first order points according to their local planarity to seed a region growing algorithm. In Ma et al. (2013), the authors find planes' normals by selecting points with the lowest curvature, and iteratively grow regions based on these points.

Finally, planes can be interpreted as sets of non-parallel lines. Holzmann et al. (2018) propose to find planes using sets of orthogonal lines that are close enough to be considered part of a same object. Their method allows them to detect less spurious planes in urban areas.

2.2 Segmentation

Region growing is one of the most prevalent segmentation approaches for the segmentation of 3D data (Cohen-Steiner et al., 2004; Whelan et al., 2015; Fang et al., 2018). Vo et al. (2015) use an adaptive octree to improve segmentation results. The RAPter algorithm (Monszpart et al., 2015), a man-made scenes reconstruction algorithm based on regular arrangements of planes, is initialized using a region growing-based over-segmentation of their input point cloud. Their algorithm is able to take into account plane primitives and inter-plane relations. Whelan et al. (2015) argue that a local curvature-based region growing algorithm can outperform RANSAC-based approaches for segmenting point clouds. Region-growing has also been coupled with a regularization step based on simple relationships between extracted primitives (coplanarity, parallelism and orthogonality) in Oesau et al. (2016).

One of the main other approaches for segmenting 3D data is the Mean-Shift algorithm (Ferraz et al., 2012). It seeks maxima of a density function, given discrete data samples. This method has the advantage of being non-parametric. Dai et al. (2018) use a mean-shift method to detect individual trees in multispectral airborne LiDAR point clouds.

Moreover, point clouds can be structured by an adjacency relationship between points, such as nearest neighbors, a triangulated mesh or a 3D Delaunay triangulation (Boissonnat, 1984). This allows us to represent the cloud as a graph, on which the vertices are the 3D points. As a benefit, efficient graph-structured clustering methods can be used, typically relying on graph-cuts (Klasing et al., 2008; Strom et al., 2010). Landrieu and Obozinski (2017) introduced the ℓ_0-cut pursuit algorithm, a greedy working-set approach to computing piecewise constant approximation of signals on such a graph. Dutta et al. (2018) adapted the Normalized Cut algorithm (Shi and Malik, 2000) to 3D LiDAR point clouds, an algorithm originally designed for perceptual grouping in raster images.

One of the main drawbacks of point cloud clustering is its tendency to either lose information on areas with high geometric complexity, or to over-segment the cloud. Thus some work focus on an non-adaptive over-segmentation to then generalize it at a given level of detail (Attene and Patanè, 2010; Lejemble et al., 2018). Nan and Wonka (2017) focus their work on an arrangement of planes from which a manifold polyhedral surface model without boundary is extracted, which can be seen as a piecewise planar approximation.

For a more complete state of the art on primitive extraction and segmentation, especially in a 3D reconstruction context, we refer the reader to Berger et al. (2017).

3. METHODOLOGY

In this paper, our objective is to compute a piecewise-planar approximation of an unstructured and noisy point cloud, acquired either from mobile mapping or terrestrial laser scanning. Even if not demonstrated here, the method should extend seamlessly to aerial or UAV liDAR scans. We denote V the set of 3D vertices in the input data. We consider that a good approximation of our data should contain as few planes as possible. As we are working in urban areas, with man-made objects, which often have simple shapes, we want to favor simple interfaces between planes as well. We use a weighted oriented graph structure $G = (V, E, w)$ to represent our data, where $E \in V \times V$ characterizes the adjacency between each point and $w \in \mathbb{R}_+^E$ stands for the weight of each edge, encoding the *closeness* between points.

Our approach can be decomposed in 2 steps: first, we compute the weighted adjacency graph G, then we compute the piecewise-planar approximation by solving a graph-structured optimization problem.

3.1 Weighted adjacency graph

If the input 3D data already has a graph structure (for instance if it is a mesh), we can simply use it as adjacency graph. If the data is an unstructured point cloud, there are several possibilities to build an adjacency graph structure (i.e. define which points are neighbors): k-nearest neighbors, triangulated meshing, 3D Delaunay triangulation, among others. We propose an adjacency graph based on the work of Guinard and Vallet (2018) which exploits sensor topology to connect the scanned points with edges and triangles and leaves isolated points unconnected. We chose to create an edge in G if and only if the two vertices belong to at least one shared triangle. Indeed, isolated points and edges of the simplicial complex correspond to parts of the scene where the geometry is not even locally surfacic, i.e. the geometry is too complex relative to scanning resolution to define a local tangent plane, such as tree foliage or linear structures for which planar approximation would not be appropriate. This acts as a prefiltering step in combination to providing the adjacency structure.

We design an edge weight which decreases with the distance between points:

$$w_{u,v} = \frac{1}{\alpha + \frac{d(u,v)}{d_0}} \;, \tag{1}$$

where d_0 is the mean length of edges in the input data, and α a non-negative constant taken here as 2.

3.2 Piecewise-planar approximation

The piecewise-planar approximation is computed by solving a graph-structured optimization problem. We use a modified version of the ℓ_0-cut pursuit algorithm of Landrieu and Obozinski (2017) which we dub *ℓ_0-plane pursuit*.

3.2.1 Formulation We denote the set of all planes of $\mathbb{R}^3$ as $\mathcal{P}$. We denote by $d(v,\pi)$ the distance between a vertex v and a plane $\pi \in \mathcal{P}$. For Π a set of planes in $\mathcal{P}^V$, we denote Π_v the plane associated to vertex v. Such plane set defines an approximation of V characterized by the projection of each vertex to its associated plane.

We aim to construct an energy $E : \mathcal{P}^V \to \mathbb{R}$ whose minimization gives a precise and simple piecewise-planar approximation of V. For $\Pi \in \mathcal{P}^V$, we propose the following energy:

$$E(\Pi) = \sum_{v \in V} d(v, \Pi_v)^2 + \mu \sum_{(u,v) \in E} w_{u,v} [\Pi_u \neq \Pi_v], \tag{2}$$

where $[\pi \neq \pi']$ is the function of $\mathcal{P}^2 \to \{0,1\}$ equal to 0 when π and π' are identical planes, and 1 otherwise. The parameter $\mu \in \mathbb{R}_+$ is the regularization strength. The first term of Equation 2 corresponds to the fidelity term of the energy. This term ensures that each point is well approximated by its corresponding plane. The second part of Equation 2 encourages the planes associated to adjacent vertices to have identical parametrization. It also forces the interface between adjacent planes to have simple shapes by penalizing edges between adjacent vertices with different planes. Therefore, a set of planes with low energy should achieve a tradeoff between shape complexity and precision of their approximation of the input point cloud. Note that our choice of w (see Equation (1)) favors cuts between distant points, insuring that vertices associated to the same plane are never too far from one another.

3.2.2 Energy minimization We define the set of approximating planes as the results of the following optimization problem:

$$\Pi^\star = \operatorname*{arg\,min}_{\Pi \in \mathcal{P}^V} E(\Pi) \;. \tag{3}$$

The energy E is non-convex, non-continuous and non-differentiable, and hence is hard to minimize. However, it is similar to the energy in the *generalized minimal partition problem* introduced by Landrieu and Obozinski (2017) to compute piecewise constant approximation of multidimensional signals on graphs. In this paper, the energy is minimized using the ℓ_0-cut pursuit algorithm. It is a working set algorithm, maintaining a partition $\mathcal{V}$ of the graph G into disjoint constant connected components. It also keeps track of the adjacency structure of the components.

The ℓ_0-cut pursuit algorithm proceeds in a top-down manner, iteratively splitting a current partition $\mathcal{V}$ in finer components. At each iteration, each component is split into two parts through a graph-cut formulation. If it is profitable with respect to the objective energy to split the component, it is replaced by the connected components of the two parts. At the end of each iteration, adjacent components whose fusion would decrease the energy are merged. The algorithm stops when no components can be further refined nor merged. While this algorithm does not guarantee to find a global optimum of the generalized minimal partition problem, in practice it is able to find good approximate solutions in a few iterations only.

Our problem differs from the generalized minimal partition problem since our variable associated with each vertex is in the space of planes instead of a multidimensional signal. However, we can follow a very similar scheme to the ℓ_0-cut pursuit algorithm to minimize E. The key steps that we adapted are:

- **Initialization:** to minimize the influence of bad initialization, we start by computing k_0 planes using the RANSAC algorithm. k_0 is chosen such that the relative decrease in energy provided by adding an extra plane is under a given threshold (chosen here to 0.005).
- **Refining:** in the ℓ_0-cut pursuit algorithm, components are split according to the optimal binary partition criterion (Landrieu and Obozinski, 2017, 3.1.1). Translated to our setting, this amounts to finding the binary partition (B, B^c) of a component U defined as follow:

  $$\operatorname*{arg\,min}_{B \subset U, (\pi,\pi') \in \mathcal{P}^2} \sum_{v \in B} d(v,\pi) + \sum_{v \in U \setminus B} d(v,\pi') + \mu \sum_{(i,j) \in E^B} w_{i,j} \;,$$

  with $E^B = E \cap (B \times B^c \cup B^c \times B)$ the set of edges linking B and $B^c = U \setminus B$. This step can be approximately solved by an alternated minimization scheme adapted from the original ℓ_0-cut pursuit algorithm. We replace the initialization step from 2-means to a 2-plane extraction step using the RANSAC algorithm. The optimization with respect to B can be performed efficiently through a graph-cut formulation, while the optimization with respect to π and π' is done through least square minimization. The scheme of this algorithm is detailed in Algorithm 1. In practice, we observe that $\texttt{ite\_split} = 3$ iterations of this scheme are sufficient.
- **Backward step**: as a greedy working-set method, the ℓ_0-cut pursuit algorithm benefits from allowing the mergeing of adjacent components as long as it decreases the global energy. This can be easily adapted to our setting: to estimate if two adjacent components should be merged, a common plane is computed and the resulting energy increase is compared to the decrease in penalty.

To summarize our algorithmic scheme, we introduce the following subroutines:

- $(\pi_1, \cdots, \pi_k) \leftarrow \texttt{RANSAC}(U, k)$: takes $U \subset V$ a set of vertices and k a number of planes as input, and returns a set of k planes extracted by the RANSAC algorithm.
- $(\pi_1, \cdots, \pi_k) \leftarrow \texttt{fit}(\mathcal{U})$: takes a set of k disjoint components $\mathcal{U} = \{U_1, \cdots, U_k\}$ as input and returns $(\pi_1, \cdots, \pi_k)$ a set of k planes fitting each vertex set U_i according to the least square criterion.

- $\mathcal{V} \leftarrow$ `associate`$(\pi_1, \cdots, \pi_k)$: takes a set of k planes as input and return a partition $\mathcal{V} = \{U_1, \cdots, U_k\}$ of V such that each component U_i contains all the points for which the plane π_i is the closest.

- $\mathcal{U}' \leftarrow$ `connected_components`$(\mathcal{U})$: takes $\mathcal{U}$ a set of disjoint components as input and return the partition of their union into connected components with respect to graph G.

- $\mathcal{U} \leftarrow$ `merge`(U, U'): takes two components U and U' adjacent in G and return their merged component if is is profitable with respect to the energy E, or $\{U, U'\}$ otherwise.

We summarize our method in Algorithm 2. The first for-loop corresponds to the split step and the second one to the merge step. We draw attention on the fact that the merge step is optional, but helps reducing the number of regions in the partition as well as decreasing the global energy.

Algorithm 1 $\mathcal{U} \leftarrow$ `split`(U)

```
$(\pi, \pi') \leftarrow \texttt{RANSAC}(U, 2)$
for ite_split iterations do
    $B \leftarrow \arg\min_{B \subset U} \sum_{v \in B} d(v, \pi) + \sum_{v \in U \setminus B} d(v, \pi') + \mu \sum_{(i,j) \in E^B} w_{i,j}$
    $(\pi, \pi') \leftarrow \texttt{fit}(\{B, U \setminus B\})$
end for
$\mathcal{U} \leftarrow \texttt{connected\_components}(\{B, U \setminus B\})$
return $\mathcal{U}$
```

Algorithm 2 ℓ_0-plane pursuit

```
$(\pi_1, \cdots, \pi_{k_0}) \leftarrow \texttt{RANSAC}(V, k_0)$          % Initialization
$\mathcal{V} \leftarrow \texttt{associate}(\pi_1, \cdots, \pi_{k_0})$
$\mathcal{V} \leftarrow \texttt{connected\_components}(\mathcal{V})$
while not_converged do
    for $U \in \mathcal{V}$ do
        $\mathcal{U} \leftarrow \texttt{split}(U)$          % Component splitting
        $\mathcal{V} \leftarrow \mathcal{V} \setminus U \cup \mathcal{U}$
    end for
    for $U, U'$ adjacent in $G$ do
        $\mathcal{U} \leftarrow \texttt{merge}(U, U')$          % Component merging
        $\mathcal{V} \leftarrow \mathcal{V} \setminus \{U, U'\} \cup \mathcal{U}$
    end for
end while
return $\mathcal{V}$, $\texttt{fit}(\mathcal{V})$
```

The ℓ_0-plane pursuit algorithm presented in Algorithm 2 returns a partition $\mathcal{V} = (U_1, \cdots, U_k)$ of V as well as a set of planes $(\pi_1, \cdots, \pi_k)$. The vertices in U_i are approximated by their projection in the plane π_i. The plane set Π such that $\Pi_v = \pi_i$ if and only if v is in U_i is an approximate minimizer of the energy E.

We want to draw attention on the fact that our formulation doesn't have any assumption on the planes' orientation and number, nor on the number of vertices per region. This allow us to produce a segmentation that is adaptive to the local geometry of the cloud, by creating more regions in complex parts of the scene while keeping a simple approximation with a small number of planes for large planar parts, such as roads or facades.

4. RESULTS

We compare our results to our own implementation of the region-growing-based method of Cohen-Steiner et al. (2004), which will serve as baseline. We do not compare our method with other state-of-the-art work such as Polyfit (Nan and Wonka, 2017) because they do not address exactly the same problem. For instance, Polyfit consider the problem of point cloud approximation for the 3D reconstruction by looking for an appropriate set of self-intersecting primitives, whereas our approach is not meant to be watertight, and each primitive can be disconnected from the other primitives. We design our metric according to our main goal: approximating a set of points with a set of planes. Our metric will be expressed as the sum of squared distances from the points to their associated plane:

$$E = \sum_{v \in \mathcal{V}} d(v, \pi_v)^2. \tag{4}$$

The method of Cohen-Steiner et al. (2004) takes a mesh and the number of desired regions $\mathcal{N}$ as input. The seed planes are chosen at random and the algorithm grows along the seeds. For every iteration of the algorithm, the set of planes $\mathcal{P}$ is recomputed. Each new plane corresponds to the best-fitting plane of each region according to the same L2 criterion that we use. The region-growing process is repeated until the algorithm converges.

This method works well for small datasets, but is not suitable for large datasets, due to its time complexity. Another major drawback of this method is the need to specify the number of regions as input: when working with a new scene, we cannot know in advance how many planes will be necessary to precisely approximate the scene.

4.1 Datasets

We evaluate our method on two urban point clouds, one outdoor and the other indoor.

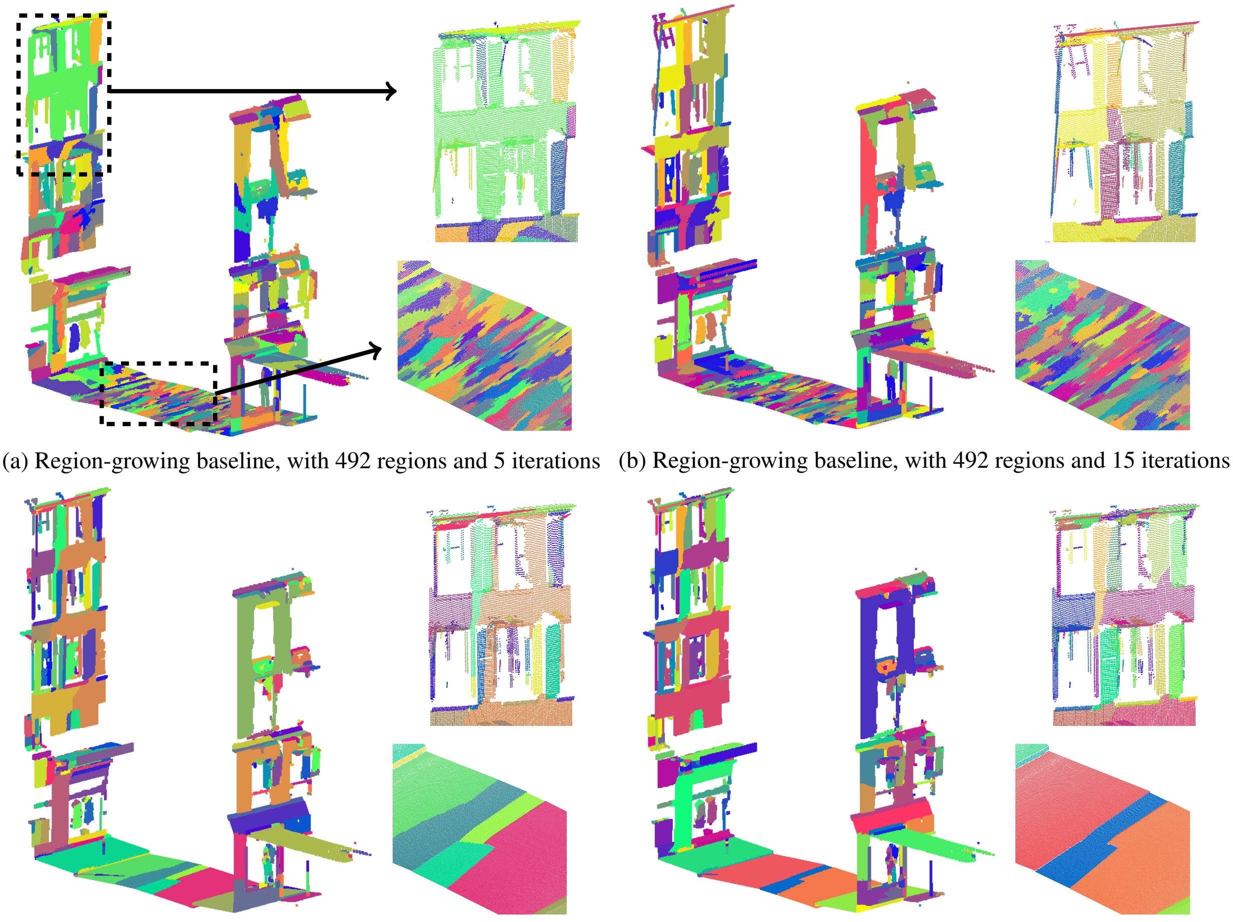
The outdoor scene was acquired via mobile mapping, with the Stereopolis vehicle (Paparoditis et al., 2012) in the streets of Paris. It contains $218,546$ points and $418,254$ triangles. The scene represents a portion of street, with a road in the middle of the scan. Two of the sides are occupied by buildings. Some ceilings are acquired through the windows present in the scan.

The indoor scene was acquired with a fixed terrestrial laser inside a small chapel by the Centre de formation ENSG-Forcalquier. The acquisition is very dense and contains $1,263,321$ points and $7,313,760$ triangles. It represents wall portions with vaults. We selected this dataset for two reasons, first to validate the scaling of our method, and then because to evaluate its precision on surfaces such as vaults which are not exactly planar. Because of the size of the data, only our method is tested on this dataset.

The clouds have been meshed with the algorithm presented in Guinard and Vallet (2018). We note that the reconstruction used is noisy and leads to the creation of isolated triangles (that may not be connected to the rest of the graph due to the use of the sensor topology). The graph structure used in our paper would create a separate sub-graph for each of such isolated triangle. Hence we cleaned the datasets to keep only the main graph components for both methods.

4.2 Numerical experiments

We also compare the baseline approach with our method on the outdoor scene. The baseline algorithm is run twice, for 5 and 15 iterations respectively. More iterations do not significantly improve the results. In order to be able to fairly compare our method



(a) Region-growing baseline, with 492 regions and 5 iterations

(b) Region-growing baseline, with 492 regions and 15 iterations

(c) Ours, without merge step, 514 regions

(d) Ours, 492 regions

Figure 2. Comparison of our method and the baseline. Each color represent a different region. Points are projected on the plane supporting their region. Each result comes with a close picture of a building and a road portion. Our method creates large regions on simple parts of the cloud, in order to be more adaptive on complex parts.

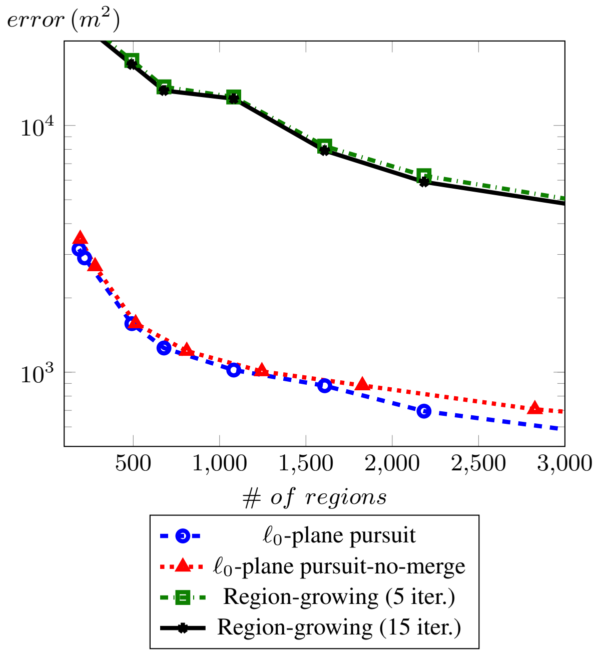


Figure 3. Comparison of the error between our method and the region-growing baseline on the urban dataset.

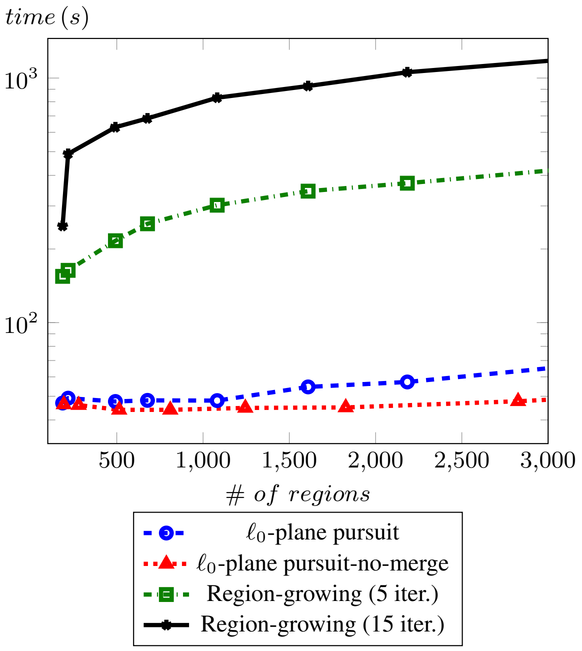
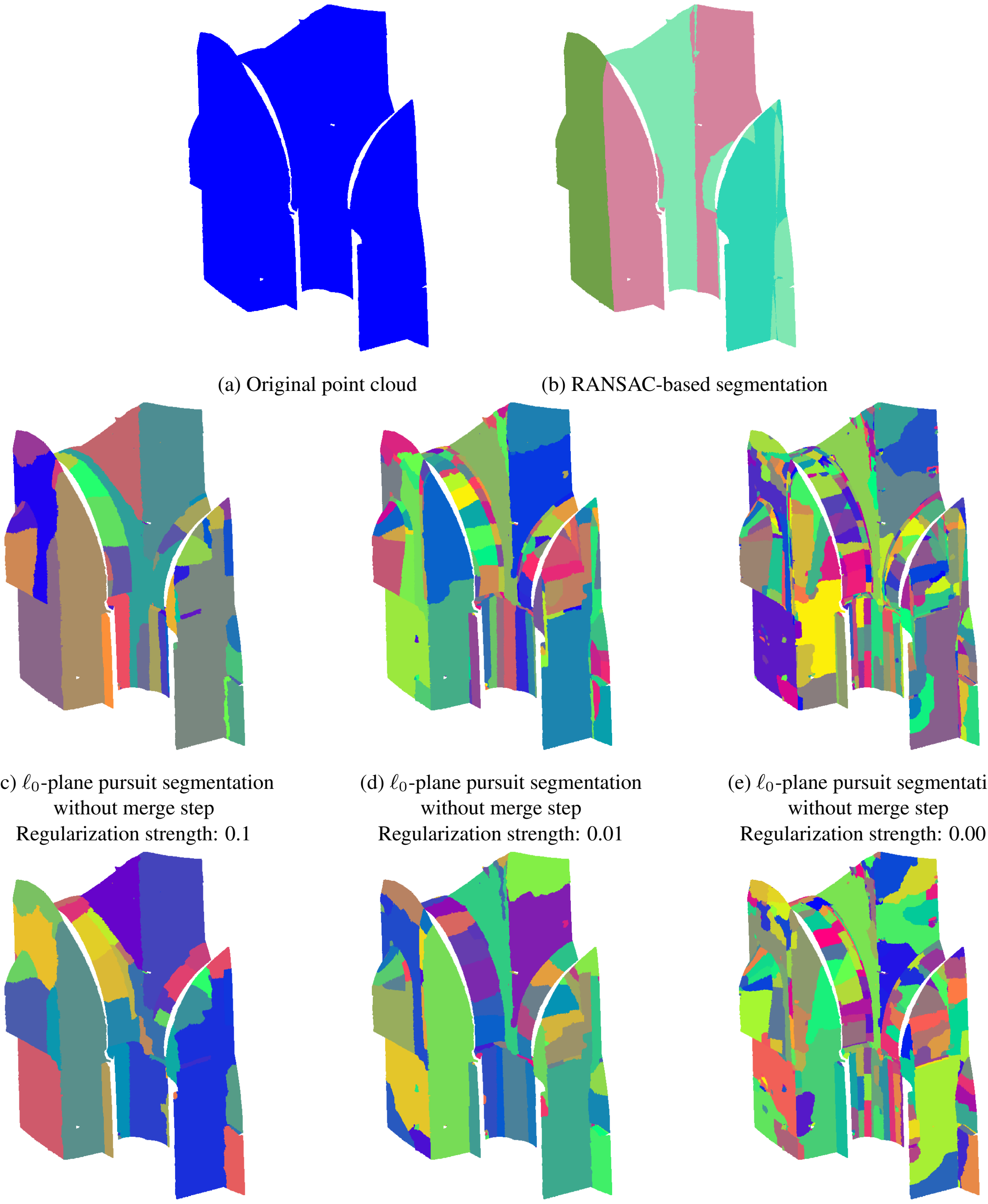


Figure 4. Comparison of the computation time in seconds between our method and the region-growing baseline.



(a) Original point cloud

(b) RANSAC-based segmentation

(c) ℓ_0-plane pursuit segmentation without merge step Regularization strength: 0.1

(d) ℓ_0-plane pursuit segmentation without merge step Regularization strength: 0.01

(e) ℓ_0-plane pursuit segmentation without merge step Regularization strength: 0.001

(f) ℓ_0-plane pursuit segmentation with merge step Regularization strength: 0.1

(g) ℓ_0-plane pursuit segmentation with merge step Regularization strength: 0.01

(h) ℓ_0-plane pursuit segmentation with merge step Regularization strength: 0.001

Figure 5. Scan of the inside of the chapel cloud, composed of walls and vaults. Each color corresponds to a different region. We remark that the use of the merge step decreases the number of regions without increasing the quality of the reconstruction.

to the baseline, we decide to evaluate both methods with the same number of clusters. Therefore, we first run our method with a given regularization strength, then we compute the baseline specifying the number of clusters found by our method. We also decided to measure the influence of the merge step in our method by launching it twice at each test: one time with the merge step (which number of clusters will be used as input parameter for our baseline) and one time without (called ℓ_0-plane pursuit-no-merge in the results).

We can observe the results on Figure 2. We see that the region-growing baseline has a tendency to cluster the data in regions of homogeneous size. This implies that simple and regular parts of the cloud, such as the road, will be clustered in more regions than on our method, decreasing the number of regions available to approximate more complex parts of the cloud. This is the main reason for the large difference in terms of performance between the baseline and our method. The detailed results for the error metric are plotted in Figure 3, and the computation times are plotted in

Figure 4.

While we tried to compare our method with the baseline on the larger indoor dataset, we could not achieve convergence with the baseline in reasonable time. The results of our method on the large indoor dataset are shown on Figure 5. Our expectations here, are that the vaults should be divided into small planar pieces elongated in the direction of minimum curvature. We observe that this is indeed what happens on Figure5. Our method is able to create large regions for large planar parts of the scene (e.g. walls) and its high adaptability allows it to create smaller regions to fit rounded parts of the scan. The detailed results are displayed on Figure 6.

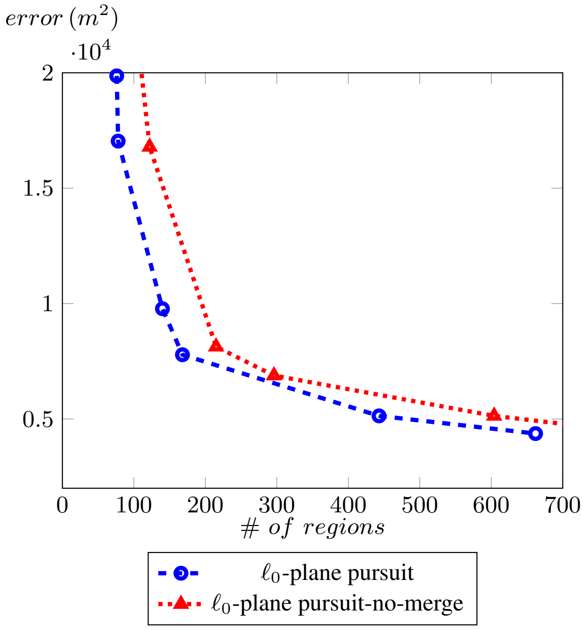


Figure 6. Comparison of the error between our method with and without the merge step on the chapel dataset.

5. CONCLUSION

In this paper, we introduced a new method for the piecewise-planar approximation of 3D data based on an adjacency structure. This method is adaptive to the local geometric complexity of the cloud and is suitable for large datasets (i.e. millions of points). Our algorithm only required a point cloud as input but can benefit from an existing adjacency information such as that provided by a triangulated mesh.

We acknowledge that our method could be improved by implementing an adaptation of the *merge-resplit* strategy presented in Landrieu and Obozinski (2017). Indeed, the merge step doesn't seem able to remove all the small artifacts occasioned by early segmentation steps.

However, the main drawback of our method is its tendency to lose the topological connection between adjacent regions. Instead, we obtain a set of planar regions that are not topologically connected. A reconstruction as in Ochmann et al. (2016) could overcome this drawback.

An interesting contribution to improve this method would be the use of multi-primitives as in Vidal et al. (2014) in our RANSAC (like spheres or cylinders) that are closer to some objects found in urban areas, such as trunks or poles. Another interesting perspective would be to use the proposed segmentation as part of a polyhedral reconstruction algorithm.

ACKNOWLEDGMENTS

The authors would like to acknowledge the DGA for their financial support of this work and Jean-Pierre Papelard (ENSG) for providing the indoor LiDAR scan.

References

Asvadi, A., Premebida, C., Peixoto, P. and Nunes, U., 2016. 3d lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. *Robotics and Autonomous Systems* 83, pp. 299–311.

Attene, M. and Patanè, G., 2010. Hierarchical structure recovery of point-sampled surfaces. In: *Computer Graphics Forum*, Vol. 29number 6, Wiley Online Library, pp. 1905–1920.

Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Guennebaud, G., Levine, J. A., Sharf, A. and Silva, C. T., 2017. A survey of surface reconstruction from point clouds. In: *Computer Graphics Forum*, Vol. 36number 1, Wiley Online Library, pp. 301–329.

Boissonnat, J.-D., 1984. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics (TOG)* 3(4), pp. 266–286.

Boulch, A., de La Gorce, M. and Marlet, R., 2014. Piecewise-planar 3d reconstruction with edge and corner regularization. In: *Computer Graphics Forum*, Vol. 33number 5, Wiley Online Library, pp. 55–64.

Chauve, A.-L., Labatut, P. and Pons, J.-P., 2010. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, pp. 1261–1268.

Cohen-Steiner, D., Alliez, P. and Desbrun, M., 2004. Variational shape approximation. In: *ACM Transactions on Graphics (ToG)*, Vol. 23number 3, ACM, pp. 905–914.

Dai, W., Yang, B., Dong, Z. and Shaker, A., 2018. A new method for 3d individual tree extraction using multispectral airborne lidar point clouds. *ISPRS journal of photogrammetry and remote sensing* 144, pp. 400–411.

Demantke, J., Mallet, C., David, N. and Vallet, B., 2011. Dimensionality based scale selection in 3d lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 38(Part 5), pp. W12.

Dutta, A., Engels, J. and Hahn, M., 2018. Segmentation of laser point clouds in urban areas by a modified normalized cut method. *IEEE transactions on pattern analysis and machine intelligence*.

Dzitsiuk, M., Sturm, J., Maier, R., Ma, L. and Cremers, D., 2017. De-noising, stabilizing and completing 3d reconstructions on-the-go using plane priors. In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, IEEE, pp. 3976–3983.

Fang, H., Lafarge, F. and Desbrun, M., 2018. Planar shape detection at structural scales. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ferraz, A., Bretar, F., Jacquemoud, S., Gonçalves, G., Pereira, L., Tomé, M. and Soares, P., 2012. 3-d mapping of a multi-layered mediterranean forest using als data. *Remote Sensing of Environment* 121, pp. 210–223.

Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6), pp. 381–395.

Guinard, S. and Vallet, B., 2018. Sensor-topology based simplicial complex reconstruction from mobile laser scanning. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*.

Holzmann, T., Maurer, M., Fraundorfer, F. and Bischof, H., 2018. Semantically aware urban 3d reconstruction with plane-based regularization. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 468–483.

Holzmann, T., Oswald, M. R., Pollefeys, M., Fraundorfer, F. and Bischof, H., 2017. Plane-based surface regularization for urban 3d reconstruction. In: *28th British Machine Vision Conference*.

Klasing, K., Wollherr, D. and Buss, M., 2008. A clustering method for efficient segmentation of 3d laser data. In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, IEEE, pp. 4043–4048.

Landrieu, L. and Obozinski, G., 2017. Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM Journal on Imaging Sciences* 10(4), pp. 1724–1766.

Lejemble, T., Mura, C., Barthe, L. and Mellado, N., 2018. Multi-scale planar segments extraction from point clouds. In: *Journées Françaises dInformatique Graphique*.

Ma, L., Favier, R., Do, L., Bondarev, E. and de With, P. H., 2013. Plane segmentation and decimation of point clouds for 3d environment reconstruction. In: *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, IEEE, pp. 43–49.

Martinovic, A., Knopp, J., Riemenschneider, H. and Van Gool, L., 2015. 3d all the way: Semantic segmentation of urban scenes from start to end in 3d. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4456–4465.

Monszpart, A., Mellado, N., Brostow, G. J. and Mitra, N. J., 2015. Rapter: rebuilding man-made scenes with regular arrangements of planes. *ACM Trans. Graph.* 34(4), pp. 103–1.

Nan, L. and Wonka, P., 2017. Polyfit: Polygonal surface reconstruction from point clouds. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy*, pp. 2353–2361.

Ochmann, S., Vock, R., Wessel, R. and Klein, R., 2016. Automatic reconstruction of parametric building models from indoor point clouds. *Computers & Graphics* 54, pp. 94–103.

Oesau, S., Lafarge, F. and Alliez, P., 2016. Planar shape detection and regularization in tandem. In: *Computer Graphics Forum*, Vol. 35number 1, Wiley Online Library, pp. 203–215.

Paparoditis, N., Papelard, J.-P., Cannelle, B., Devaux, A., Soheilian, B., David, N. and Houzay, E., 2012. Stereopolis ii: A multi-purpose and multi-sensor 3d mobile mapping system for street visualisation and 3d metrology. *Revue française de photogrammétrie et de télédétection* 200(1), pp. 69–79.

Peternell, M. and Steiner, T., 2004. Reconstruction of piecewise planar objects from point clouds. *Computer-Aided Design* 36(4), pp. 333–342.

Schnabel, R., Wahl, R. and Klein, R., 2007. Efficient ransac for point-cloud shape detection. In: *Computer graphics forum*, Vol. 26number 2, Wiley Online Library, pp. 214–226.

Shi, J. and Malik, J., 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence* 22(8), pp. 888–905.

Strom, J., Richardson, A. and Olson, E., 2010. Graph-based segmentation for colored 3d laser point clouds. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, IEEE, pp. 2131–2136.

Tatavarti, A., Papadakis, J. and Willis, A. R., 2017. Towards real-time segmentation of 3d point cloud data into local planar regions. In: *SoutheastCon, 2017*, IEEE, pp. 1–6.

Vidal, V., Wolf, C. and Dupont, F., 2014. Mechanical Mesh Segmentation and Global 3D Shape Extraction. PhD thesis, Université Lyon 1-Claude Bernard; INSA Lyon.

Vo, A.-V., Truong-Hong, L., Laefer, D. F. and Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing* 104, pp. 88–100.

Whelan, T., Ma, L., Bondarev, E., de With, P. and McDonald, J., 2015. Incremental and batch planar simplification of dense point cloud maps. *Robotics and Autonomous Systems* 69, pp. 3–14.

Xie, H., Wang, J., Hua, J., Qin, H. and Kaufman, A., 2003. Piecewise c1 continuous surface reconstruction of noisy point clouds via local implicit quadric regression. In: *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, IEEE Computer Society, p. 13.

Xu, B., Jiang, W., Shan, J., Zhang, J. and Li, L., 2015. Investigation on the weighted ransac approaches for building roof plane segmentation from lidar point clouds. *Remote Sensing* 8(1), pp. 5.