

# SCALABLE AND DETAIL-PRESERVING GROUND SURFACE RECONSTRUCTION FROM LARGE 3D POINT CLOUDS ACQUIRED BY MOBILE MAPPING SYSTEMS

Daniela Craciun<sup>a,\*</sup>, Andrés Serna Morales<sup>b</sup>, Jean-Emmanuel Deschaud<sup>a</sup>, Beatriz Marcotegui<sup>b</sup> and François Goulette<sup>a</sup>

MINES ParisTech, PSL - Research University

<sup>a</sup> CAOR - Centre de Robotique, 60. Bd. St Michel, 75006 Paris, France

<sup>b</sup> CMM - Centre de Morphologie Mathématique, 35 rue St Honoré, 77300 Fontainebleau, France

**KEY WORDS:** surface, reconstruction, LiDAR, geometry, urban, mobile mapping.

## ABSTRACT:

The currently existing mobile mapping systems equipped with active 3D sensors allow to acquire the environment with high sampling rates at high vehicle velocities. While providing an effective solution for environment sensing over large scale distances, such acquisition provides only a discrete representation of the geometry. Thus, a continuous map of the underlying surface must be built. Mobile acquisition introduces several constraints for the state-of-the-art surface reconstruction algorithms. Smoothing becomes a difficult task for recovering sharp depth features while avoiding mesh shrinkage. In addition, interpolation-based techniques are not suitable for noisy datasets acquired by Mobile Laser Scanning (MLS) systems. Furthermore, scalability is a major concern for enabling real-time rendering over large scale distances while preserving geometric details. This paper presents a fully automatic ground surface reconstruction framework capable to deal with the aforementioned constraints. The proposed method exploits the quasi-flat geometry of the ground throughout a morphological segmentation algorithm. Then, a planar Delaunay triangulation is applied in order to reconstruct the ground surface. A smoothing procedure eliminates high frequency peaks, while preserving geometric details in order to provide a regular ground surface. Finally, a decimation step is applied in order to cope with scalability constraints over large scale distances. Experimental results on real data acquired in large urban environments are presented and a performance evaluation with respect to ground truth measurements demonstrate the effectiveness of our method.

## 1. INTRODUCTION AND MOTIVATION

Generating continuous 3D models of urban environments, at ground level, is becoming an increasing need for a wide range of applications. Driving simulation engines, semantic perception for intelligent vehicles, trajectory planning and visual servoing for autonomous navigation are several applications requiring an automatic framework for producing accurate and scalable 3D models over large scale distances.

Mobile Mapping Systems (MMS) equipped with active 3D sensors are well adapted for acquiring dense 3D measurements of the underlying surface, while driving in normal traffic conditions. Nevertheless, such a discrete representation must be further exploited in order to build a continuous surface via 3D modeling techniques. In addition, 3D modeling from MMS datasets requires scaling-up cartography which includes processing big datasets in an automatic fashion, designing consistent zoom levels for multi-scale mapping but also cloud computing for massive scalability of geo-specific 3D model databases.

In this research work we focus on the ground surface reconstruction theme which in dense urban environments is concerned with the road, sidewalk and ramp access areas. These sharp depth changes and geometrical details need to be preserved in order to cope with the accuracy required by the visual layer of simulator engines or for intelligent vehicles.

When dealing with the surface reconstruction problem using 3D point clouds acquired by MLS sensors, several key issues must be addressed. Noise sources coming from the laser sensing device, external calibration and mobile acquisition (distance to the scanned surface, incidence angle, surface geometry and material type) must be carefully identified and modeled correspondingly.

Smoothing the noise in MLS data while preserving sharpness is still a difficult task for the state-of-the-art surface reconstruction algorithms. Furthermore, scalability issues must be addressed in order to deal with real-time rendering of big datasets acquired over large distances. In this paper we propose a fully automatic surface reconstruction framework for ground areas which copes with the aforementioned constraints imposed by MLS datasets. Our paper is organized as follows: Section 2 presents the existing solutions for ground surface reconstruction from 3D point clouds. Section 3 provides an overview of our framework which is driven by a ground segmentation module presented in Section 4. The ground points are exploited along with a novel surface reconstruction pipeline described in Section 5. Section 6 evaluates the performances of the proposed framework, while Section 7 presents quantitative results obtained over large scale distances. Section 8 summarizes the work and presents future extensions of our method.

## 2. RELATED AND PROPOSED RESEARCH WORK

Ground surface is one of the main components of man-made environments; it corresponds either to road and sidewalks in outdoor urban scenes, or to the floor in indoor environments. At the global level, the ground surface reconstruction usually consists in two steps: ground extraction and surface reconstruction. Special attention must be given when combining these steps in order to enable scaling-up cartography. In addition, the core of the framework relies on the surface reconstruction method which must be well chosen in order to recover sharp depth features and to preserve geometric details while dealing with scalability aspects. This section presents an overview of the existing ground modeling systems and reviews the available surface reconstruction techniques, while focusing on detail-preserving and scalability requirements.

\*Corresponding author: daniela.craciun@mines-paristech.fr

**Ground modeling systems.** A very important step in the ground surface reconstruction process is represented by its extraction from the global scene. In (Carlberg et al., 2008), the surface reconstruction of a scan segment is preceded by a ground extraction step and combined with a mesh segmentation procedure based on a proximity criterion applied over the triangles. In (Wiemann et al., 2010) the polygonal map is exploited to perform floor, ceiling and wall labeling by clustering planar regions with similar attributes based on region growing principle. However, in cases where the reconstruction of the entire scene is not required, these approaches result in extra-computation for both steps, reconstructing non-ground objects and for eliminating them.

In this paper, the ground surface reconstruction workflow starts by separating the ground from non-ground objects in order to apply a planar surface reconstruction algorithm to the ground area. Beside the computational effectiveness (reconstruct only the ground), this approach leaves the possibility to integrate geometrically consistent surface reconstruction algorithms (adapted to complex or non-planar objects).

Existing methods (Carlberg et al., 2008) and (Jaakkola et al., 2008) reported automatic mesh-based reconstruction frameworks for road modeling. However, these techniques do not recover sharp depth features commonly present in man-made environments (such as roads, sidewalks and ramp access) and they do not address scalability issues.

We propose an automatic framework which combines an automatic segmentation with a high-detailed surface reconstruction framework capable to preserve sharp depth features and to generate a regular surface, while dealing with scalability issues.

**Surface reconstruction methods.** Surface reconstruction methods can be roughly categorized in two major classes. The first class is represented by implicit techniques (Alexa et al., 2003), (Otzireli et al., 2009) which aim at estimating new mesh vertices from the original acquisition by performing a polynomial fitting over a local planar neighbourhood. The implicit function is usually exploited with the Marching Cubes algorithm (Lorenson and Cline, 1987) to compute the surface. Methods for computing watertight meshes were also designed based on implicit functions, such as Poisson method (Kazhdan et al., 2006). The aforementioned methods have several drawbacks with respect to our concerns. They require normal computation which, beside the computationally expensiveness, they rely on locally planar approximations which may not hold for mobile acquisition at varying vehicle speed. Moreover, implicit methods introduce new mesh vertices, while reducing the amount of the original 3D points. Consequently, they do not guarantee geometry preservation with respect to the point cloud which represents the closest approximation of the scanned surface. The second class is represented by explicit methods (Edelsbrunner, 1998), (Gopi et al., 2000) which proceed by triangulating the 3D points based on different geometric criteria, providing a reconstructed surface very close to the point cloud. In (Carlberg et al., 2008), authors introduce a surface reconstruction method which proceeds by triangulating series of scan lines, taking advantage of the acquisition setup. In (Marton et al., 2009), authors report a greedy triangulation performed within a region growing approach. Nevertheless, these methods do not deal with noise for recovering sharp depth features and they do not address scalability issues for enabling a detail-preserving surface reconstruction framework capable to process automatically big datasets acquired over large distances. Surface reconstruction algorithms based on interpolation were also reported (Soucy and Laurendeau, 1995), (Bernardini et al., 1999). Nevertheless, interpolation-based methods do not cope

with noisy datasets as those acquired by MMS. In (Digne et al., 2014), authors introduce a volumetric approach. These methods require massive simplification (via edge and vertices collapsing operations) which is very ambiguous, leading to loss of detail and artifacts.

In our research work we are interested in generating a triangular representation which is as close as possible to the scanned surface. To this end, it is important to keep the original 3D point cloud as the mesh vertices. For this reason, our attention is directed towards the use of an explicit surface reconstruction method which does not rely on normal computation and exploits the quasi-flat geometry of the ground generally verified in structured outdoor environments. We also compare the results of our method with respect to two representative frameworks belonging to each class: Poisson (Kazhdan et al., 2006) and Greedy (Marton et al., 2009) methods.

**Scalability.** The research work reported in this paper addresses the geometrical scalability problem which aims at reducing redundant data, while preserving geometric details. Clustering planar regions has been an alternative for building primitive-based 3D models (Wiemann et al., 2010). On the other hand, vertex removal has also been reported (Carlberg et al., 2008) to reduce redundant data coming from multi-date laser scans. Nevertheless, these approaches do not recover nor preserve geometric details, providing more coarse 3D modeling solutions.

This paper presents an automated ground surface reconstruction framework which provides a regular surface, recovers sharp depth features and deals with scalability aspects. The proposed method comes together with a parallel scheme dedicated to massive 3D point cloud processing.

### 3. OVERVIEW OF THE PROPOSED METHOD

The overall processing workflow of the proposed surface reconstruction method comes together with the global framework illustrated in Figure 1 which has as input a massive 3D point cloud acquired by a MMS. The dataset is first sliced into 3D chunks of  $N$  Mpts (Million points) each, where  $N$  denotes the number of 3D measurements recorded per chunk. According to the vehicle speed, the length of the surveyed area may vary. Figure 2 (a) illustrates an example of a 3D chunk acquired over a dense urban area situated in Paris, France. We do not make any assumption about the acquisition setup, so the input data can be supplied by different platforms (Paparoditis et al., 2012), (Goulette et al., 2006).

When choosing a surface reconstruction method, the geometric properties of the underlying surface must be taken into account in order to design an adaptive framework, geometrically consistent with each object. To this end, the proposed surface reconstruction framework starts with an automatic ground extraction phase performed through the use of a 3D point cloud segmentation and classification algorithm (Serna and Marcotegui, 2014) which assigns semantic labels with respect to different classes: ground, buildings, urban furniture and cars.

Please note that the ground surface reconstruction algorithm described in this paper can also receive as input a ground area obtained by different segmentation methods (Rusu and Cousins, 2011), such as planar segmentation through the use of RANSAC technique (Fischler and Bolles, 1981).

In a second step, the 3D points corresponding to the ground are injected into the surface reconstruction procedure which combines a planar Delaunay triangulation method with smoothing

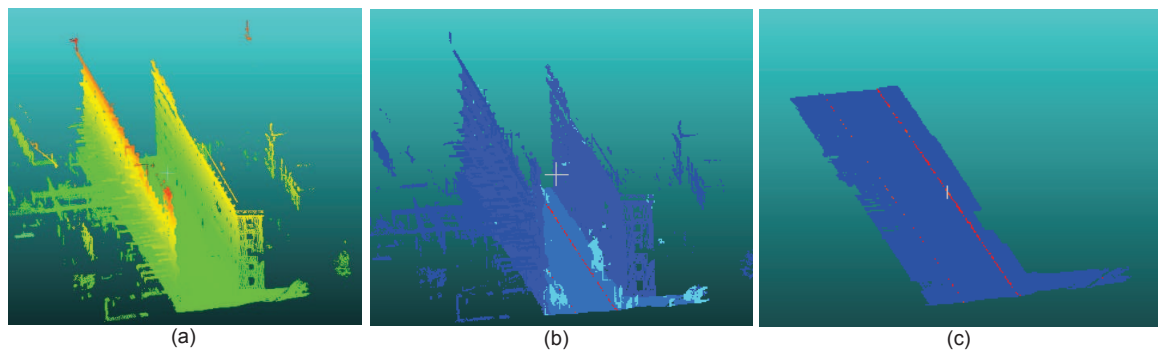


Figure 2: Dataset Urban #1: (a) Example of 3D chunk containing  $N = 3$  Mpts, acquired in Paris (France) over an approximative length of 82 m. For visualization purposes, the original point cloud is colored with respect to elevation values; (b) 3D object segmentation results: facades - dark blue, road and sidewalks - blue, non-ground objects - light blue, sidewalk borders and ramp access - red; (c) the extracted ground corresponding to road, sidewalks and ramp access: 1.27 Mpts.

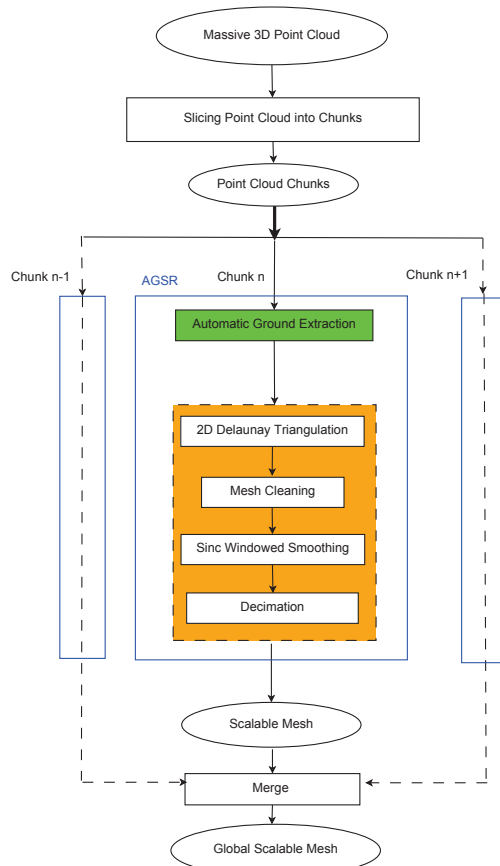


Figure 1: The workflow of the proposed Automatic Ground Surface Reconstruction (AGSR) framework designed to be integrated within a global parallel computation scheme dedicated to massive 3D point cloud processing.

and decimation techniques to generate automatically a regular and scalable ground mesh representation. Each decimated mesh resulted from the surface reconstruction method is further merged within a global reference scene. This paper is concerned with the automatic ground surface reconstruction (AGSR) phases, mainly its extraction and surface reconstruction which are described in the following two sections.

#### 4. 3D POINT CLOUD SEGMENTATION AND CLASSIFICATION

The focus here is the accurate and automatic segmentation of 3D point clouds from MLS data, applying the method proposed in (Serna and Marcotegui, 2014). It is based on elevation images and it uses image processing techniques, specially Mathematical Morphology (Matheron, 1975, Serra, 1988). The general workflow is composed by several steps: first, the 3D point cloud is projected to an elevation image. At that point, ground is segmented and object hypotheses are generated as discontinuities on the model. Then, small and isolated regions are eliminated. Facades are segmented as the highest vertical structures. Finally, the segmented image is reprojected to the 3D point cloud in order to get the final result. The following two sections are dedicated to a detailed description of ground segmentation and classification steps. For further details and complete analysis in each step, the reader is encouraged to review (Serna and Marcotegui, 2014).

##### 4.1 Elevation Images

Elevation images are 2.5D structures that contain altitude information at each pixel. They can be processed quickly, implicitly define neighborhood relationships and require less memory than 3D data. Elevation images are generated by an orthographic projection of the 3D point cloud using a virtual camera. This projection is a transformation from  $\mathbf{R}^3 \rightarrow \mathbf{N}^2$ . The virtual camera is located on the horizontal plane with normal vector  $\vec{n} = (0, 0, 1)$  and crossing the lowest point in the point cloud  $(0, 0, z_{min})$ . Thus, each pixel on the elevation image contains the vertical distance from the projected point to the camera plane. The only free parameter of this projection is the pixel size ( $pw$ ). This parameter is directly related with the resolution and the computational cost of the method. In our experiments, we have found a trade off using square pixels of  $5\text{cm} \times 5\text{cm}$  in the spatial domain.

As several points may be projected on the same pixel, we store the maximal elevation, the minimal elevation and the number of points (called accumulation) projected on the same pixel.

After images creation, a morphological interpolation is performed in order to fill holes caused by occlusions and missing scan lines. An interpolation technique based on the morphological fill holes  $\text{Fill}(f)$  is used since this transformation does not create new regional maxima in the image.

## 4.2 Ground Extraction

Ground segmentation is a critical step since urban objects are assumed to be located on it. When objects are filtered out it is possible to define the digital terrain model. With the aim of segmenting the ground, we use the approach proposed in (Hernández and Marcotegui, 2009). It is based on the  $\lambda$ -flat zones labeling algorithm defined in (Meyer, 1998) as following: let  $f$  be a digital gray-scale image  $f : D \rightarrow V$ , with  $D \subset Z^2$  the image domain and  $V = [0, \dots, R]$  the set of gray levels. Two neighboring pixels  $p, q$  belong to the same  $\lambda$ -flat zone of  $f$ , if their difference  $|f_p - f_q|$  is smaller than or equal to a given  $\lambda$  value. For all  $x \in D$ , let  $A_x(\lambda)$  be the  $\lambda$ -flat zone of image  $f$  containing pixel  $x$ . Using this definition, it is possible to obtain the ground  $g = \max \arg\{|(A_x(\lambda))|\}$  as the largest  $\lambda$ -flat zone in the elevation image. The parameter  $\lambda$  is set to 20 cm because it is usually high enough to join road and sidewalk without merging other objects, even if there are not ramp access for the sidewalk. Figure 2 (b) presents the segmentation and classification result corresponding to the input point cloud illustrated in Figure 2 (a). Figure 2 (c) depicts the 3D point cloud representing the ground composed by roads, sidewalks and accessibility ramps. The 3D points belonging to the ground are further injected into the surface reconstruction process which is described in the following section.

## 5. GROUND SURFACE RECONSTRUCTION

The ground surface reconstruction module transforms a 3D point cloud previously labelled as ground (illustrated in Figure 2 (c)), into a continuous and scalable surface representation. The proposed framework is composed by several steps which are illustrated in Figure 1 and described through the following sections. First, the 3D point cloud representing the ground is triangulated in the  $(x, y)$  plane using a constraint Delaunay algorithm which provides points connectivity. Then, we apply a mesh cleaning process to eliminate long triangles. In order to provide a continuous and regular surface model of the road, we apply the Sinc Windowed (Taubin et al., 1996) smoothing algorithm which eliminates high frequencies, while preserving sharp depth features and avoiding surface shrinkage. In a final step, a progressive decimator (Schroeder et al., 1992), (Hoppe, 1996) is applied to the smoothed mesh in order to cope with scalability constraints when performing surface reconstruction over large scale distances. It provides surface representation with low memory usage, enabling efficient data transmission and visualization. In addition, the decimation procedure enables progressive rendering in order to deal with real-time constraints imposed by driving simulation engines.

### 5.1 Point Cloud Triangulation

Let us note with  $\mathbf{P} = \{x_i, y_i, z_i | i = 1, \dots, N_p\}$  the 3D point cloud corresponding to the ground, where  $N_p$  denotes the number of points. We apply the Triangle algorithm (Shewchuk, 1996) to the 3D point cloud  $\mathbf{P}$  to generate a planar constraint Delaunay triangulation with angles no smaller than  $30^\circ$ . Let us note with  $\mathcal{M}_{DT}$  the resulting ground mesh, which has  $N_t \approx 2N_p$  triangles.

### 5.2 Long Triangles Elimination

In order to eliminate long triangles from non-uniform boundary points, we perform statistics on the edge lengths and identify those with maximum length, noted  $e_{max}$ . We identified that long edges correspond to  $e_{max} \approx \delta \bar{e}$ , where  $\bar{e}$  denotes the mean length computed over all edges  $e_i^j$  belonging to the mesh  $\mathcal{M}_{DT}$ , i.e. over all triangles  $t^j \in \mathcal{M}_{DT}, j = 1, \dots, N_t$  and for its corresponding edges  $e_i^j, i = \{1, 2, 3\}$ . The term  $\delta$  denotes a proportionality factor. A triangle  $t^j$  is eliminated if any of its edges  $e_i^j > e_{max}, i = \{1, 2, 3\}$ . This criterion ensures that only long triangles belonging to the boundary are eliminated; moreover, since small triangles are not eliminated, holes can not be generated within the mesh. In practice, for several datasets acquired in urban areas by different MLS systems (Paparoditis et al., 2012), (Goulette et al., 2006), we found that a coefficient  $\delta = 20$  results in a mesh without long triangles, which we note  $\mathcal{M}_C$ .

### 5.3 Building a Regular Surface

As illustrated in Figures 3 (a) and (b), the triangulation of noisy 3D measurements results in high frequency peaks. Since we want to inject the ground surface model in driving simulator engines, an important issue which needs to be addressed is the geometrical accuracy. The 3D model must be distortion-free and regular. In order to obtain a regular surface, the Sinc windowed smoothing procedure (Taubin et al., 1996) is applied which approximates low-pass filters by polyhedrons in order to eliminate high frequency peaks. Figures 3 (c) and (d) illustrate the smoothed mesh, noted  $\mathcal{M}_S$ ; it can be observed that the Sinc Windowed smoothing technique provides a regular surface, while preserving roads and sidewalk borders sharpness.

### 5.4 Scalability

The smoothed mesh has a high number of triangles, being redundant and causing a high memory usage. Moreover, in order to merge several mesh segments into a global scene, the mesh resolution must be drastically reduced. To this end, we apply the progressive decimation method (Schroeder et al., 1992), (Hoppe, 1996), mainly the default implementation available in the VTK library (Schroeder et al., 2006). The mesh resolution  $r(\mathcal{M}_D)$  is controlled by the reduction factor, noted  $f_D(\%)$ .

The algorithm proceeds as follows: first, each vertex is classified and inserted in a priority queue for further processing. The priority is set in order to minimize the distance to the original mesh caused by the vertex elimination and by the re-triangulation of the resulting hole. As stated in (Schroeder et al., 1992), following the vertex type (simple, interior, boundary, etc.), a different distance criterion is computed (distance to plane, distance to edge). Let us note with  $\mathcal{M}_D$  the decimated mesh, with  $N_t^D$  the corresponding number of triangles.

Figure 4 illustrates the result obtained for the input depicted in Figure 2 (c) reducing  $f_D = 90\%$  of the entire mesh. The remaining number of triangles corresponds to  $r(\mathcal{M}_D) = 10\%$  of the original mesh.

It can be observed that the decimation algorithm preserves the reconstruction of the road, sidewalk borders and accessibility ramps. In order to emphasize the detail-preserving capability of the decimation algorithm, Figure 5 illustrate the speed bump reconstruction after applying a maximal mesh reduction factor of  $f_D = 90\%$ .

**Accuracy of the decimated mesh.** As in (Turnet and Zakhori, 2013), we evaluate the accuracy of the decimated mesh by measuring the distance between the original point cloud  $\mathbf{P}$  and the



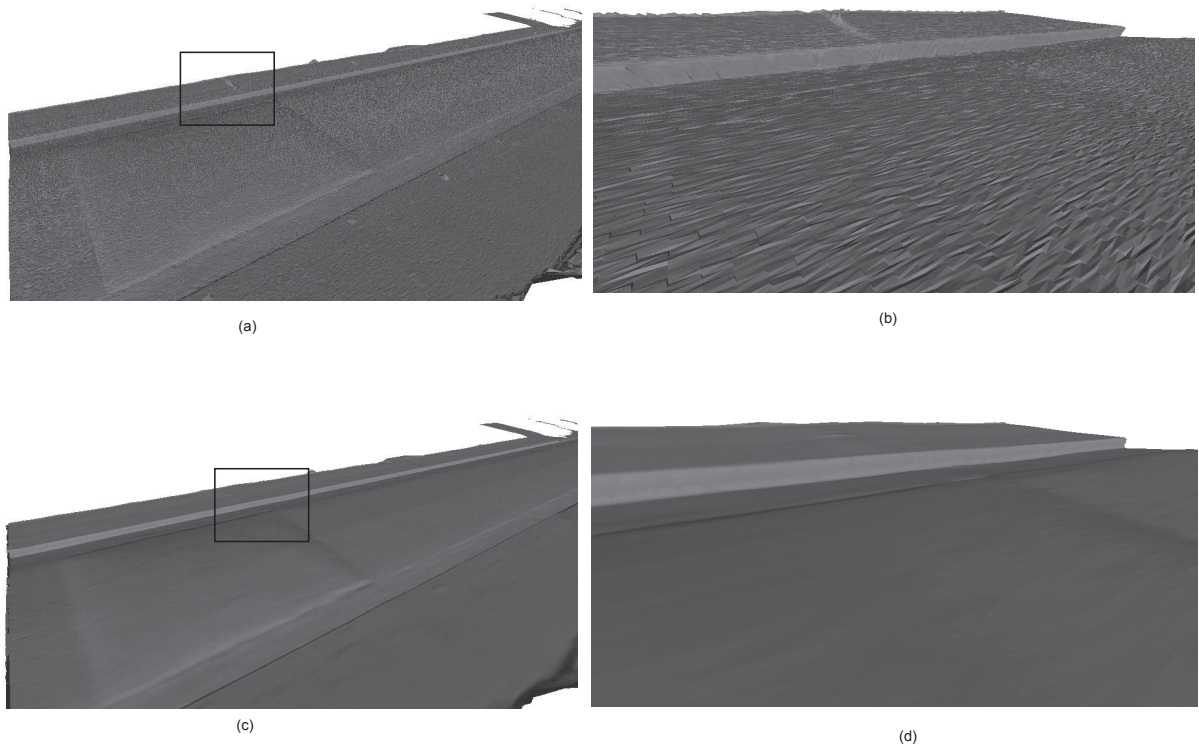


Figure 3: Mesh smoothing results on dataset Urban #2. (a) Delaunay triangulation; (b) zoom-in view on the sidewalk border corresponding to the black rectangle area illustrated in Figure (b); (c) the result of the Sinc windowed smoothing procedure (d) zoom-in view on the sidewalk border corresponding to the black rectangle area illustrated in Figure (c).

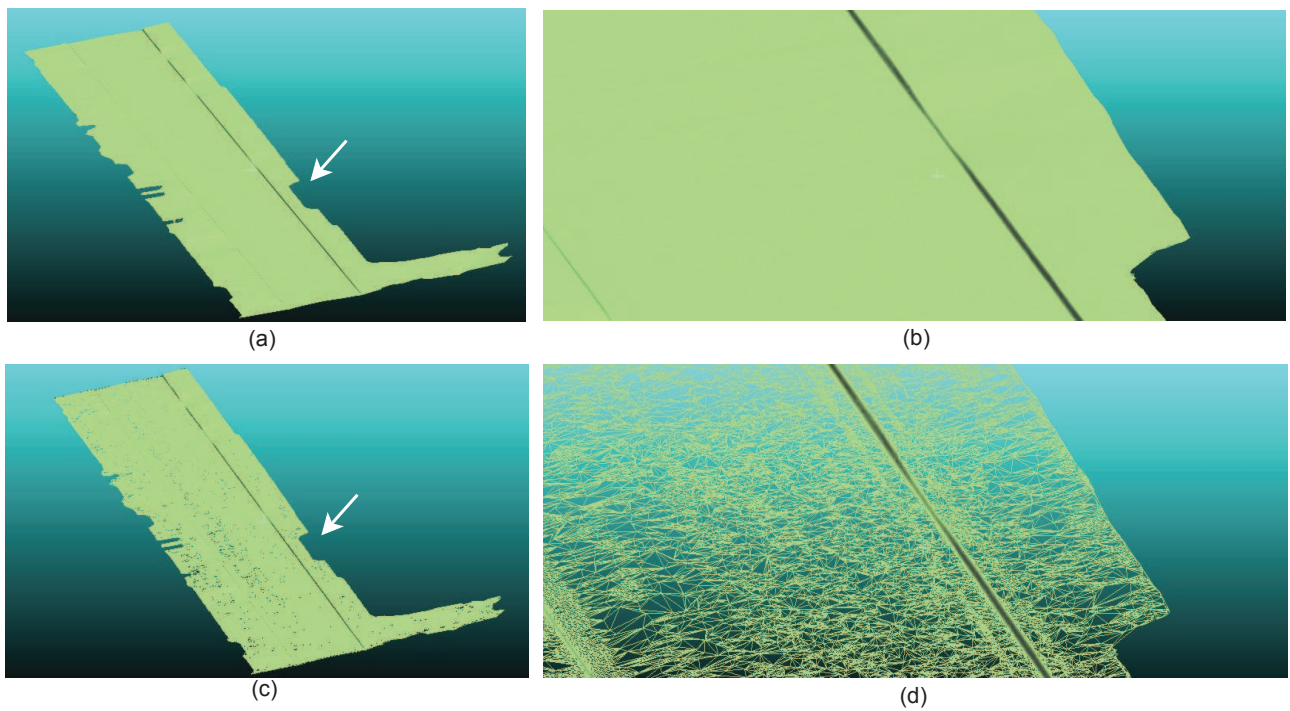


Figure 4: Decimation results obtained for the dataset Urban #1: (a) full resolution smoothed mesh:  $N_t = 2.54$  Mpts; (b) zoom-in view in the area around the sidewalk border indicated by the white arrow from Figure (a); (c) wire-frame view of the decimated mesh:  $f_D = 90\%$ ,  $N_t^D : 254$  kTriangles, (d) wire-frame view: zoom-in view in the area around the sidewalk border indicated by the white arrow from Figure (c).

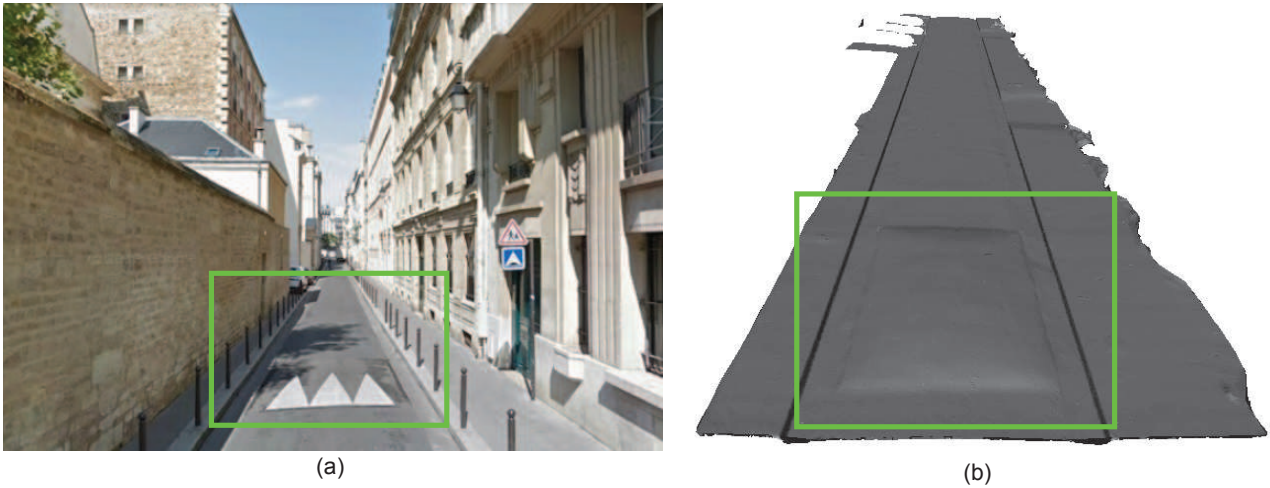


Figure 5: Surface reconstruction results for the dataset Urban #2 over 51 m length. (a) Google street view of the surveyed area (although not exactly the same pose, it is useful for surface inspection purposes), (b)  $N_p = 1.01$  Mpts,  $N_t = 2.026$  MTriangles, decimated mesh with  $f_D = 90\%$ ,  $N_t^D = 203$  kTriangles.

vertices of the decimated mesh,  $\mathcal{M}_D$ . We choose to compute the Hausdorff distance (Cignoni et al., 1998), we study both, the mean and the root mean squared ( $RMS_H$ ) distance for different mesh resolutions  $r(\mathcal{M}_D)$ . We observed that the mean is less sensible to the decimation process, while the  $RMS_H$  varies with a higher amplitude, although negligible ( $\pm 10^{-3}$  m). This let us conclude that the memory usage can be reduced by a maximal factor of  $f_D = 90\%$ , without sacrificing the accuracy of the model.

## 6. PERFORMANCE EVALUATION

We evaluate the performances of the proposed framework in terms of accuracy, memory usage and computation time.

**Accuracy evaluation.** We quantify the accuracy of the reconstructed surface with respect to several ground truth measurements which were performed manually on site (Cassette road, situated in Paris, France), mainly: the height of the sidewalk border, the height of the access and the road width, noted  $H_{sidewalk}$ ,  $H_{ramp}$  and  $W_{road}$ , respectively. Table 1 illustrates the ground truth and the reconstructed dimensions for dataset Urban #2 illustrated in Figure 5. It can be observed the reachable accuracy is better than 1.5 cm.

Dataset Urban #2	$H_{sidewalk}$	$H_{ramp}$	$W_{road}$
GT	10.5 (cm)	2.5 (cm)	3.5 (m)
Reconstruction	10.1 (cm)	2.3 (cm)	3.514 (m)

Table 1: Accuracy evaluation of the ground surface reconstruction with respect to ground truth (GT) data for Urban #2.

**Computation time.** We evaluate our algorithm on a 64b Linux machine, equipped with 32 Gb of RAM memory and an Intel Core i7 running at 3.40 GHz. Our method is implemented in C/C++ and exploits PCL (Rusu and Cousins, 2011) and VTK (Schroeder et al., 2006) libraries. Table 2 illustrates the computation time obtained for the dataset Urban #2. We can observe that the decimation step is the most expensive phase, being related to the decimation factor  $f_D$ . In this example, a maximum decimator factor was used  $f_D = 90\%$  for a mesh with 2 MTriangles, which results in 9 sec of computation time.

**Memory usage.** Table 3 illustrates the memory usage for each surface reconstruction step. It can be observed that the mesh rep-

Steps	$P_S$	$\mathcal{M}_{DT}$	$\mathcal{M}_C$	$\mathcal{M}_S$	$\mathcal{M}_D$
CPU (s)	2	2.14	0.18	3	9

Table 2: Computation time for dataset Urban #2 illustrated in Figure 5 where  $P_S$  denotes the point cloud segmentation phase for the ground extraction; each column gives the runtime corresponding to each output of the algorithm. The overall computation time is about 17s for  $N_p=1.01$  Mpts and  $N_t=203$  kTriangles.

resentation is more efficient than the point-based one, allowing to reduce the memory usage 3 times for the full resolution mesh and 20 times for a resolution mesh of  $r(\mathcal{M}_S) = 10\%$ . These results show that the proposed surface reconstruction framework provides a memory efficient surface representation, while preserving geometric details.

**Visual rendering.** The frame frequency, measured in frames per second (FPS), allows to quantify the quality of a 3D model with respect to the visual rendering capability. The second row of Table 3 illustrates the frame frequency, noted  $\nu_{rate}$  and measured using Cloud Compare (Girardeau-Montaut, 2013) for different surface representations (discrete and continuous). It can be observed that the point-based representation detains faster rendering capabilities than the full resolution mesh, which does not cope with real-time rendering requirements. In contrast, the decimated mesh exhibits real-time frame rates, while providing a continuous surface representation.

Urban #2	<b>P</b>	$\mathcal{M}_{DT}$	$\mathcal{M}_S$	$\mathcal{M}_D$
Memory (Mb)	14.856	81.61	37.600	3.7
$\nu_{rate}$ (FPS)	267.74	10.273	12.448	131.96

Table 3: Memory usage and frame frequency measures corresponding to the input chunk **P** and to the main outputs of the algorithm for dataset Urban #2 illustrated in Figure 5.

Although the decimation step is the most computationally expensive processing block of the proposed surface reconstruction framework, it enables real-time rendering of a continuous surface over large scale scenes, while preserving geometric details.

**Ground surface comparison.** We evaluate the results of the proposed framework, entitled Automatic Ground Surface Reconstruction (AGSR), with two well known surface reconstruction techniques. The first method is based on implicit functions (Kazhdan et al., 2006), while the second is an explicit method (Marton

et al., 2009) which proceeds by a greedy projection. Figure 6 and Table 4 illustrate the results obtained by applying each reconstruction algorithm to the point cloud  $\mathbf{P}$  associated to the ground area depicted in Figure 2 (c).

Dataset Urban #1	Poisson	Greedy	AGSR
$N_p^{out}$	748 k	1.28 M	127 k
$N_t^{out}$	1.5 M	2.764 M	254 k
CPU (s)	133	422	26

Table 4: Comparison between surface reconstruction methods: results obtained by running the algorithms on the dataset Urban #1.  $N_p^{out}$  and  $N_t^{out}$  denote the number of vertices and the number of triangles corresponding to each output, respectively.

By visually inspecting Figures 6 (a) and 6 (d), it can be observed that although Poisson method provides a watertight surface, it results in mesh shrinkage around the sidewalk borders. Moreover, it reduces the number of points considerably, introducing thus inaccuracies between the point cloud geometry and the final surface. In contrast, the Greedy projection method keeps all the measurements provided by the acquisition. Nevertheless, it results in discontinuity and high frequency peaks. The third row of Table 4 illustrates the computation time obtained using PCL implementations. It can be observed that the proposed method increases the performances not only in terms of accuracy, as showed in Figure 6, but also in terms of computation time. More precisely, it allows to decrease the runtime 5 times when compared to Poisson method, and 16 times with respect to the Greedy projection technique. Both methods, Poisson and Greedy, are computationally expensive due to the normal computation step. When comparing final results, it can be observed that, although the proposed technique includes a computationally expensive decimation phase, beside the detail-preserving rendering capability, it features real-time surface reconstruction on parallel processing units.

## 7. SCALING-UP DETAIL-PRESERVING GROUND MODELS

We test the surface reconstruction algorithm on several chunks acquired over urban areas acquired with different MLS systems (Paparoditis et al., 2012), (Goulette et al., 2006); different smoothing parameters were applied for each MLS. Figure 7 illustrates the result obtained on several chunks corresponding to dataset Urban #2 acquired by a MLS equipped with a Riegl sensor (Paparoditis et al., 2012). Figure 7 shows the results obtained for 4 scans segments acquired over 217 m, containing 12 Mpts, from which 4 Mpts were classified as belonging to ground. It can be observed that due to the vehicle speed which may vary, the acquired 3D chunks have different lengths. By taking into account the computation time obtained for dataset Urban #2, in average, we process 3 Mpts for 50 m length of surveyed area in about 17 s. For 100 Mpts (100 scan segments with 30% ground), it is possible to obtain the ground surface in about 28 min. For 100 km, the ground surface can be computed in about 10h. Even though time scalability is not our prior concern, by increasing the computational resources by a factor of 10, we provide a surface reconstruction framework capable to perform in real time. In this upgraded configuration, the algorithm can deliver the reconstruction of the entire road network for a country with 10000 km length in about 5 days, non-stop driving and data acquisition at 90 km/h.

## 8. CONCLUSIONS AND FUTURE WORK

This paper presented Automatic Ground Surface Reconstruction (AGSR), a fully automatic framework for generating detail-

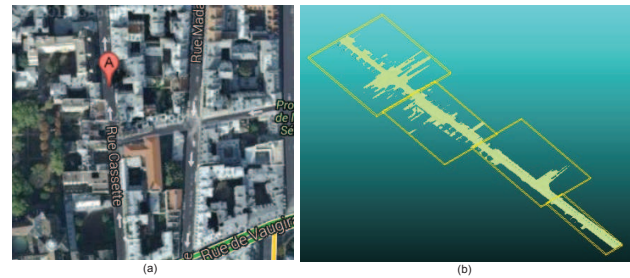


Figure 7: Surface reconstruction results obtained for dataset Urban #2. (a) Google street view of the surveyed area, (b) surface reconstruction results obtained for 4 chunks with different lengths (each one included in its bounding box), overall approximative distance: 217 m.

preserving and scalable ground surface reconstruction from MLS data in outdoor environments. The presented research work is mainly concerned with urban areas, where the ground is usually represented by quasi-flat zones; for off-road terrain modeling, adapted schemes must be designed. The proposed algorithm comes with a parallel scheme to be expanded at two levels: i) for running on series of ground chunks, and ii) at upper level, for reconstructing different classes, allowing to integrate adapted surface reconstruction frameworks for complex (non-planar) objects. The proposed method emphasizes the high potential of the MMS which, when combined with suitable frameworks, it allows to generate accurate and scalable 3D models in a fully automatic fashion. Future work focuses on the facade surface reconstruction step, while tacking into account the integration of multiple mesh surfaces (ground and facades) within a global referential frame.

## REFERENCES

- Alexa, M., Behr, J., Cohen-or, D., Fleishman, S., Levin, D. and Silva, C. T., 2003. Computing and rendering point set surfaces. *IEEE Transactions on Computer Visualization and Graphics* 9, pp. 3–15.
- Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C. and Taubin, G., 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 5, pp. 349–359.
- Carlberg, M., Andrews, J., Gao, P. and Zakhov, A., 2008. Fast surface reconstruction and segmentation with ground-based and airborne lidar range data. *International Conference 3D PVT*.
- Cignoni, P., Rocchini, C. and Scopigno, R., 1998. Measuring error on simplified surfaces. *Computer Graphics Forum* 17, pp. 167–174.
- Digne, J., Cohen-Steiner, D., Alliez, P., Desbrun, M. and de Goes, F., 2014. Feature-preserving surface reconstruction and simplification from defect-laden point sets. *Journal of Mathematical Imaging and Vision* 48, pp. 369–382.
- Edelsbrunner, H., 1998. Shape reconstruction with Delaunay complex. *Proceedings of the Third Latin American Symposium on Theoretical Informatics*, Springer-Verlag 1, pp. 119–132.
- Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24(6), pp. 381–395.
- Girardeau-Montaut, D., 2013. Telecom ParisTech and EDF. Cloud Compare. <http://www.danielgm.net/cc/>.



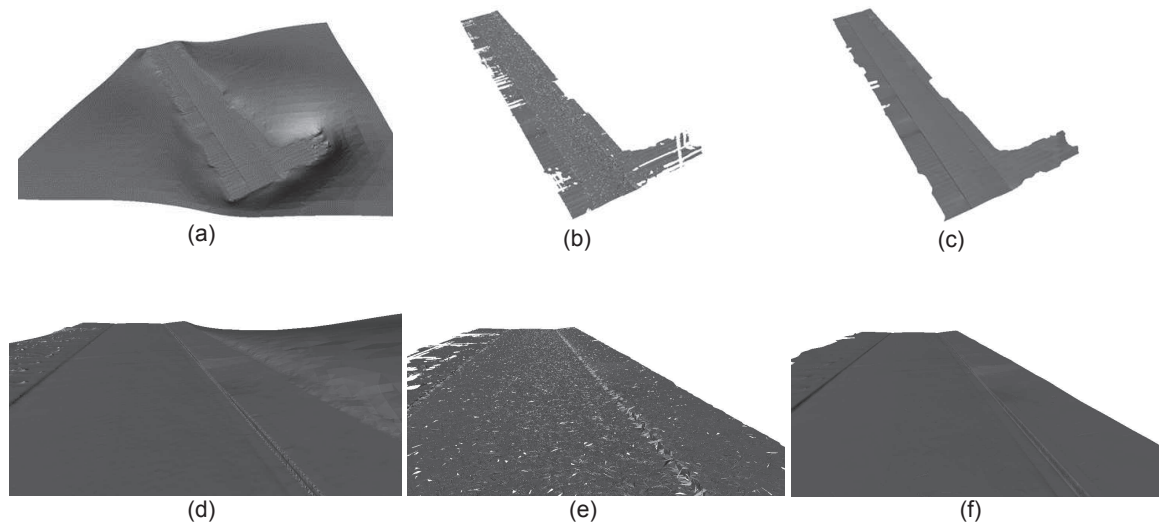


Figure 6: Comparison between surface reconstruction methods obtained for the dataset Urban #1: (a) Poisson method, (b) Greedy projection method, (c) **AGSR** method; (d) - (f): close-up views corresponding to the surface reconstructions results illustrated in Figures (a) - (c).

Gopi, M., Krishnan, S. and Silva, C. T., 2000. Surface reconstruction based on lower dimensional localized Delaunay triangulation. *Eurographics*.

Goulette, F., Nashashibi, F., Abuhadrous, I., Ammoun, S. and Laurgeau, C., 2006. An integrated on-board laser range sensing system for on the way city and road modelling. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Hernández, J. and Marcotegui, B., 2009. Filtering of artifacts and pavement segmentation from mobile lidar data. In *ISPRS workshop Laser scanning, XXXVIII-3/W8 2*, pp. 329–333.

Hoppe, H., 1996. Progressive meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH* pp. 99–108.

Jaakkola, A., Hyypä, J., Hyypä, H. and Kukko, A., 2008. Retrieval algorithms for road surface modelling using laser-based mobile mapping. In *Sensors, Open Access* pp. 5238–5249.

Kazhdan, M., Bolitho, M. and Hoppe, H., 2006. Poisson surface reconstruction. *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* pp. 61–70.

Lorensen, W. E. and Cline, H. E., 1987. Marching Cubes: A high resolution 3D surface construction algorithm. *Computer Graphics* 21, pp. 163–169.

Marton, Z. C., Rusu, R. B. and Beetz, M., 2009. On fast surface reconstruction methods for large and noisy point clouds. In *IEEE International Conference on Robotics and Automation* pp. 3218–3223.

Matheron, G., 1975. *Random sets and integral geometry*. John Wiley & Sons, New York 28, pp. 493–501.

Meyer, F., 1998. *From connected operators to levelings. Mathematical Morphology and its Applications to Image and Signal Processing*, Kluwer Academic Publishers 12, pp. 191–198.

Otzireli, A. C., Guennebaud, G. and Gross, M., 2009. Feature preserving point set surfaces based on non-linear kernel regression. *Eurographics, Guest Editor P. Dutre and M. Stamminger* 28, pp. 493–501.

Papadoditis, N., Papelard, J., Cannelle, B., Devaux, A., Soheil-ian, B., David, N. and Houzay, E., 2012. Stereopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualization and 3D metrology. *Revue Française de Photogrammétrie et de Télédétection* pp. 69–79.

Rusu, R. B. and Cousins, S., 2011. 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation (ICRA)*.

Schroeder, W. J., Zarge, J. A. and Lorensen, W. E., 1992. Decimation of triangle meshes. In *Conference Proceedings of SIGGRAPH* pp. 65–70.

Schroeder, W., Martin, K. and Lorensen, B., 2006. *The visualization toolkit. Third Edition* Kitware Inc.

Serna, A. and Marcotegui, B., 2014. Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. *In Press* 28, pp. 493–501.

Serra, J., 1988. *Image analysis and mathematical morphology: Theoretical advance*. Academic Press 2, pp. 493–501.

Shewchuk, J. R., 1996. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In *Applied Computational Geometry: Towards Geometric Engineering* (Ming C. Lin and Dinesh Manocha, editors), Springer-Verlag (Berlin) 1148, pp. 203–222.

Soucy, M. and Laurendeau, D., 1995. A general surface approach to the generation of a set of range views. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17, pp. 334–358.

Taubin, G., Zhang, T. and Golub, G., 1996. Optimal surface smoothing as filter design. *IBM Research Report* 1148, pp. 203–222.

Turner, E. and Zakhor, A., 2013. Watertight planar surface meshing of indoor points-clouds with voxel carving. In *IEEE International Conference on 3D Vision* pp. 41–48.

Wiemann, T., Nuchter, A., Lingemann, K., Stiene, S. and Hertzberg, J., 2010. Automatic construction of polygonal maps from point cloud data. *International Workshop on Safety, Security and Rescue Robotics (SSRR '10)*.