HAL
open science

Automatic Ground Surface Reconstruction from mobile laser systems for driving simulation engines

Daniela Craciun, Jean-Emmanuel Deschaud, Francois Goulette

To cite this version:

Daniela Craciun, Jean-Emmanuel Deschaud, Francois Goulette. Automatic Ground Surface Reconstruction from mobile laser systems for driving simulation engines. SIMULATION: Transactions of The Society for Modeling and Simulation International, 2017, 93, pp.201-211. hal-01474498

HAL Id: hal-01474498

https://hal.science/hal-01474498v1

Submitted on 22 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noname manuscript No.
(will be inserted by the editor)

Automatic Ground Surface Reconstruction from Mobile Laser Systems for Driving Simulation Engines

Daniela Craciun · Jean-Emmanuel Deschaud* · Francois Goulette*

Special Issue Driving Simulation

Abstract Driving simulation engines represent a cost effective solution for vehicle development, being employed for performing feasibility studies, tests failure and for assessing new functionalities. Nevertheless, they require geometrically accurate and realistic 3D models in order to allow drivers training. This paper presents the Automatic Ground Surface Reconstruction (AGSR) method, a framework which exploits 3D data acquired by Mobile Laser Scanning (MLS) systems. They are particularly attractive due to their fast acquisition at terrestrial level. Nevertheless, such a mobile acquisition introduces several constraints for the existing 3D surface reconstruction algorithms. The proposed surface modeling framework produces a regular surface and recovers sharp depth features within a scalable and detail-preserving framework. Experimental results on real data acquired in urban environments allow us to conclude on the effectiveness of the proposed method.

Keywords: surface reconstruction · LiDAR · driving simulator engines · road network · mobile laser systems

1 Introduction

Driving simulation engines require geometrically accurate and realistic 3D models of urban environments. Nowadays, such 3D models are computed manually by graphic designers who combine a wide variety of data ranging from GPS car maps to aerial images, passing through GIS data [1]. However, the resulted 3D models lack geometrical accuracy and photorealism, limiting therefore drivers training in real conditions. A more difficult task is represented by the road modeling process as it requires very accurate geometrical information in order to supply drivers perception for cars maneuverability. In order to overcome the limitations of the existing 3D road modeling methods, several research

*Corresponding author

MINES ParisTech, PSL Research University, Centre for robotics, 60 Bd St Michel 75006 Paris, France
Tel.: 0033(0)1 40 51 94 54
Fax: 0033(0)1 43 26 10 51
E-mail: firstname.lastname@mines-paristech.fr

Manuscript received : 14 December 2014, revised : 19 June 2016, accepted : 17 October 2016

projects [2] are directed towards the use of MLS systems which allow sensing the environment of their surroundings with high sampling rates at high vehicle velocities. MLS systems provide geometrically accurate 3D measurements at terrestrial level over large scale distances. Nevertheless, such mobile acquisition results in a high amount of data which requires a fully automatic road surface reconstruction framework.

When dealing with the surface reconstruction problem using 3D point clouds acquired by MLS, several key issues must be addressed, such as ensuring scalability while preserving sharp depth changes and geometrical details, often sensible to smoothing operations.

The research work reported in this paper aims at exploiting 3D data acquired by a MLS system for automatic generation of geometrically accurate surface reconstruction in urban environments for driving simulation engines. In this paper we propose a fully automatic surface reconstruction framework for roads and sidewalks which copes with the aforementioned constraints imposed by MLS systems, while fulfilling the requirements of driving simulation engines.

The paper is organized as following. Section 2 introduces our method for improving perceptive realism from 3D data acquired by MLS systems and the implementation of ground 3D models within the simulator software. The next section presents the existing solutions for ground surface reconstruction from 3D point clouds acquired by MLS. Section 5 provides an overview of our framework which is driven by a ground segmentation module presented in Section 6. The ground points are exploited along with a novel surface reconstruction pipeline described in Section 7. Section 8 evaluates the performances of the proposed framework, while Section 9 presents quantitative results obtained over large scale distances. Section 10 summarizes the obtained results and presents future extensions of our method.

2 Perceptive Realism from 3D Point Clouds acquired by MLS Systems

Driving simulation engines represent a cost effective alternative for improving vehicle development with minimum costs. Such systems allow the simulation of a wide variety of traffic scenarios with visually enriched environments for develop- ing vehicle dynamics, driving assistance systems and car lighting.

Perceptive realism from scanned reality. Driving simulation engines fuse visual, audio and motion senses within a global architecture composed by several modules. A detailed description of the functional structure of a simulator engine can be found in [3]. The spatio-temporal coherence in a driving simulation engine is a major concern. It is related to the proprioceptive integration, i. e.: humans' sensibility to delay and perception incoherence (depth, motion) [4], [5]. If they are not treated accordingly, they can lead to severe misperception, headaches and accidents. A major concern in car manufacturing is represented by the use of realistic data and driving scenarios for designing adapted functional units. This requires consistent resources for collecting real-time traffic information such as vibrations, visual data bases, sounds and traffic incidents. As presented in [6], realistic restitution of longitudinal and lateral acceleration improves realism during driving simulation. A critical component in generating a suitable visual layer for driving simulation

engines is represented by the realism of the 3D model which must be correlated to both, cars vibrations [7] and sound component.

Visual layer from MLS data. The use of GIS (Geographic Information Systems) data within driving simulator engines provides an effective testbed for vehicle developing. The visual layer is composed by two main ingredients: 3D environment models and the road network supplied by GIS datasets. Nowadays, such 3D models are created by graphic designers through the use of manual frameworks. In presence of occlusions, missing data is filled with synthetic information extracted from similar non-occluded areas. Such workflows do not provide a real model, producing drivers' misperception. In addition, continuous changing in urban planning requires up-to-date 3D models and GIS datasets. This calls for automatic procedures capable to survey and generate 3D models over large distances in a relatively short time. Furthermore, the cost for generating manually 3D models represents in average a third of the overall expenses required by a driving simulation engine.

From MLS data to scalable road networks via logical description. In order to overcome the fastidious processing of manual methods, the design of automated 3D modeling frameworks becomes a must. In addition, with the new advancements in mobile mapping systems, it is now possible to acquire real data, at terrestrial level while driving in normal traffic conditions. This allows acquiring real data and generating 3D models over large distances within a cost effective methodology. Nevertheless, such a mobile acquisition results in a high amount of data which requires automated 3D modeling frameworks.

The workflow presented is paper was developed within an ongoing research project, [2] which is focusing on the generation of geo-specific 3D models for driving simulation engines in order to allow vehicle design and drivers' training with minimal costs. The project is mainly concerned with the design of an automatic framework capable to generate geometrically accurate 3D models from MLS data acquired over large distances. The reconstructed ground surfaces generated by our algorithm are further exploited via a logical description for road networks encoded in different file formats, such as CityGML [8] or RoadXML [9], accepted by driving simulation engines. They are widely employed to supply the software of driving simulation engines. A good example is SCANeR^{TM} [10] which provides a complete description of roads network for a variety of driving simulation engines.

3 Open Issues for Ground Surface Reconstruction from MLS datasets

Mobile mapping systems (MMS) equipped with active 3D sensors are well adapted for acquiring densely sampled 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 through means of 3D modeling.

The existing surface reconstruction systems have reached a maturity level when dealing with stop-and-go mapping systems. However, when the input data is a 3D point cloud delivered by the latest mobile mapping systems, new key issues must be addressed in order to cope with several constraints such as mobile acquisition, scalability and detail-preserving capabilities required for the surface reconstruction algorithms.

The mobile acquisition introduces new challenges to the existing surface reconstruction algorithms. They are represented by internal and external calibration steps, the accuracy of the sensor localisation and other parameters related to the acquisition (distance to the scanned surface, incident angle, surface geometry, etc.) which must be carefully identified and modeled correspondingly.

MMS must be embedded with 3D modeling frameworks capable to scale-up over large distances, while preserving geometrical details such as sharp features and depth changes. This is required by the capacity to process big data sets in a fully automatic fashion and to design consistent level of detail (LOD) for multi-resolution mapping of geo-specific 3D model data bases. In addition, scalability issues must be addressed in order to deal with real-time rendering of big data sets acquired over large distances.

This paper is concerned with the ground surface reconstruction problem, which in man-made outdoors environments corresponds to the road, sidewalk and ramp access areas. These are structured areas, including sharp depth changes and geometrical details, which need to be preserved in order to cope with the accuracy required by the visual layer of driving simulation engines. This requires noise smoothing procedures able to deal with MLS datasets in order to eliminate noise while preserving sharpness. This is an open issue which must be addressed in order to provide a highly accurate surface of road borders, ramp access and other geometric details.

This paper presents a fully automatic algorithm designed for supplying ground surface reconstruction from MLS datasets. The proposed framework addresses the aforementioned constraints, being able to preserve geometric details, while being scalable over large distances.

4 State-of-the-Art on Ground Surface Reconstruction Systems

Existing systems on ground surface reconstruction can be classified with respect to ground modeling and surface reconstruction algorithms. This section reviews main methods belonging to each class.

Ground modeling systems proceed either by first building a mesh and then segmenting the ground from the mesh based on different criterions, or by first extracting the ground from the point cloud and then reconstruct the surface. In [11], the ground is segmented from the reconstructed mesh based on a proximity criterion applied over the triangles. In [12], authors propose a ground modeling procedure for indoor environments which allows floor, wall and ceiling segmentation based on planar clustering procedure. The aforementioned methods proceed by meshing the entire point cloud, resulting in extra-time computation for reconstructing the non-ground objects and for eliminating them. When only the ground is required, it is more efficient to first extract the ground and then to build the mesh.

The ground surface reconstruction workflow presented in this paper separates first the ground from the non-ground objects and proceeds by reconstructing the 3D point cloud belonging to the ground. This allows to apply adapted surface reconstruction frameworks, consistent with planar (road, floor) or non-planar (complex) objects.

The state of the art on ground modeling systems can be roughly classified in semi-automatic [13] and automatic frameworks. In [13], authors introduce a high-resolution surface reconstruction algorithm for road areas selected by human operator. While improving the state of the art with a highly accurate ground reconstruction method, the proposed framework is not adapted for automatic processing over large scale distances. Mesh-based methods reported in [11], [14] present the advantage of being automatic and thus adapted for on-line data acquisition and processing over large distances. Nevertheless, several key issues of the existing techniques stand in the capability to reconstruct sharp depth features and geometrical details, while being scalable over long distances.

The second main processing block required by a ground modeling system is represented by surface reconstruction method for which reported frameworks can be classified in two major classes: implicit and explicit methods. The first class of algorithms [15], [16] proceed by polynomial fitting, while the second class [17], [18] reconstruct the surface by triangulating directly the 3D points, resulting in a surface very close to the acquired 3D point cloud.

In this research work we are interested in generating a triangular mesh as close as possible to the scanned surface and thus, an explicit method is more adapted to our application. The results obtained are compared to two well known surface reconstruction methods belonging to each class: Poisson [19] and Greedy projection [20] techniques.

5 Overview of the Automatic Ground Surface Reconstruction (AGSR) Algorithm

The proposed framework does not exploit any assumption for the acquisition setup, being therefore suitable for 3D point clouds acquired by a large variety of MMS [21]. The algorithm presented in this paper was tested on various datasets supplied by two different MLS systems which are illustrated in Figure 1. The proposed ground surface reconstruction algorithm comes together with a parallel scheme designed for supplying massive 3D point cloud processing acquired by MLS systems. Figure 2 depicts the global architecture of the AGSR method.

The main input is a massive 3D point cloud which is first sliced into 3D chunks, with N Mpts (Million points) per chunk. The length of a 3D chunk varies with the vehicle speed. In Figure 3 (a) it can be observed an example of a 3D chunk extracted from a survey performed over an urban area located in Paris, France.

In this research work, the surface reconstruction method exploits a 3D point cloud segmentation and classification algorithm [22] for the ground extraction phase. This procedure assigns semantic labels to each 3D measurement, providing a classification output with different classes: ground (composed by the road, sidewalks and ramp access), buildings, urban furniture and cars. Such a semantic labeling scheme provides two advantages: (i) it gives the possibility to parallelize the surface reconstruction at class level, while adapting the surface reconstruction method with respect to its geometric properties; (ii) in dynamic environments, when similar objects are detected, the already computed model can be inserted within a global reference scene.

(a) (b)

Fig. 1 Mobile mapping systems employed in the present research work: (a) L3D2 MMS prototyped by the Robotic Lab of the Mines ParisTech [30], equipped with a Velodyne 3D sensing device, (b) STEREOPOLIS prototype designed by the French Mapping Agency [29].

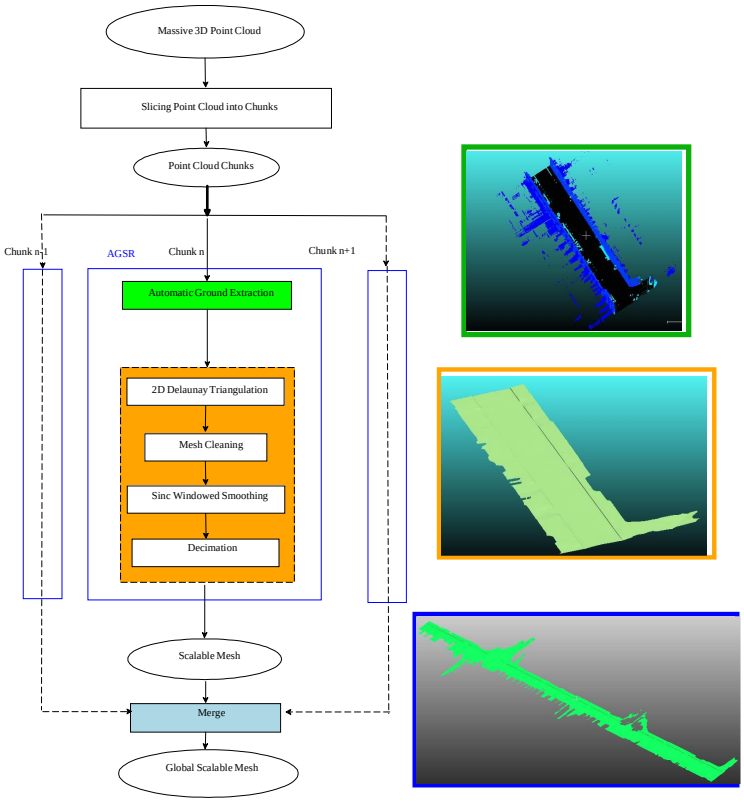


Fig. 2 The global architecture of the AGSR method and its integration within a parallel computation scheme dedicated to massive 3D point cloud processing. On the right side are illustrated the outputs corresponding to three procedures composing the workflow: ground extraction - green, surface reconstruction - orange, merged decimated meshes - blue.

The second phase of the algorithm exploits the 3D point cloud corresponding to the ground to build a triangular mesh. The algorithm starts by a planar Delaunay triangulation process which is followed by a smoothing phase in order to reduce noise, generating thus a regular and drivable ground surface. In order to cope with scalability issues over large scale distances, a decimation stage is applied to the

smoothed mesh. In a final step, a global referential frame is updated with each mesh corresponding to each 3D chunk. The proposed workflow is designed to be applied in parallel to each 3D chunk. The following two sections are dedicated to a detailed description of the two main processing blocks of the AGSR framework: the automatic ground extraction step and the surface reconstruction phase.

6 Automatic Ground Extraction

This section describes ground extraction procedure which represents the first step in the 3D modeling process. In this work we employ the segmentation algorithm proposed in [22]. It exploits elevation images along with Mathematical Morphology [23] [24] tools. The algorithm is composed of three processing blocks: the first one is dedicated to the projection of the 3D point cloud onto an elevation image. In a second step, ground and object segmentation is processed by analyzing discontinuities over the elevation images. Facades segmentation is performed by identifying highest vertical structures. The final procedure back-projects each pixel of the elevation image to the 3D space. Each 3D point is labelled, allowing to recover 3D points corresponding to the ground.

Figure 3 (b) presents the segmentation and classification result corresponding to the input point cloud depicted in Figure 3 (a). Figure 3 (c) depicts the 3D point cloud representing the ground composed by roads, sidewalks and accessibility ramps. For a complete description of the 3D point cloud classification procedure, the reader may refer to [22]. The 3D points belonging to the ground are identified with respect to the labeling provided by the classification procedure and exploited along with the surface reconstruction process which is described in the following section.

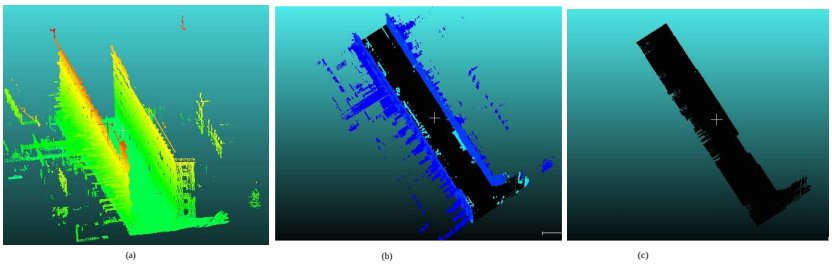


Fig. 3 Ground extraction results: (a) example of 3D chunk acquired over Assas road located in Paris (France): approximative length 82m, with N = 3 Mpts, color coded with respect to elevation values, (b) 3D point cloud segmentation and classification results: facades - dark blue, road, sidewalks and ramp access - black, non-ground objects - light blue, (c) the 3D point cloud corresponding to the ground: 1.27 Mpts.

7 Automatic, Scalable and Detail-Preserving Ground Surface Reconstruction

The ground surface reconstruction module transforms a 3D point cloud labelled as ground (illustrated in Figure 3 (c)), into a continuous and scalable surface representation. The proposed framework is composed by several steps which are illustrated in Figure 2 and described through the following sections. First, the 3D point cloud representing the ground is triangulated in the (x, y) plane using a Delaunay triangulation algorithm [25] 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 smoothing algorithm [26] which eliminates high frequencies, while preserving sharp depth features and avoiding surface shrinkage. In a final step, a progressive decimator [2 7] , [28] is applied to the smoothed mesh in order to cope with scalability constraints when performing surface reconstruction over large scale distances. The decimation phase 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.

7.1 Point Cloud Triangulation

Let us note with $\mathbf{P} = \{x_i, y_i, z_i | i = 1, .., N_p\}$ the 3D point cloud corresponding to the ground, where N_p denotes the number of points. We apply the Triangle algorithm [25] to the 3D point cloud $\mathbf{P}$ to generate a planar constraint Delaunay triangulation with angles no smaller than 30°. Let us note with M_{DT} the resulting ground mesh, which has $N_t \approx 2N_p$ triangles.

7.2 Long Triangles Elimination

In order to eliminate long triangles from non-uniform boundary points, we perform statistics over 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^j belonging to the mesh M_{DT}, i.e. over all triangles $t^j \in M_{DT}$, $j = 1, .., N_t$ and for its corresponding edges e^j, $i = \{1, 2, 3\}$. The term δ denotes a proportionality factor. A triangle t^j is eliminated if any of its edges $e^j > e_{max}$, $i = \{1, 2, 3\}$. 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 [29], [30], we found that a coefficient $\delta = 20$ results in a cleaned mesh, i.e. without long triangles, which we note M_C.

7.3 Building a Regular Surface

As illustrated in Figures 4 (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 [26] is applied which approximates low-pass filters by polyhedrons in order to eliminate high frequency peaks. Figures 4 (c) and (d) illustrate the resulted smoothed mesh, noted M_S; it can be observed that the Sinc Windowed smoothing technique provides a regular surface, while preserving roads and sidewalk borders sharpness.

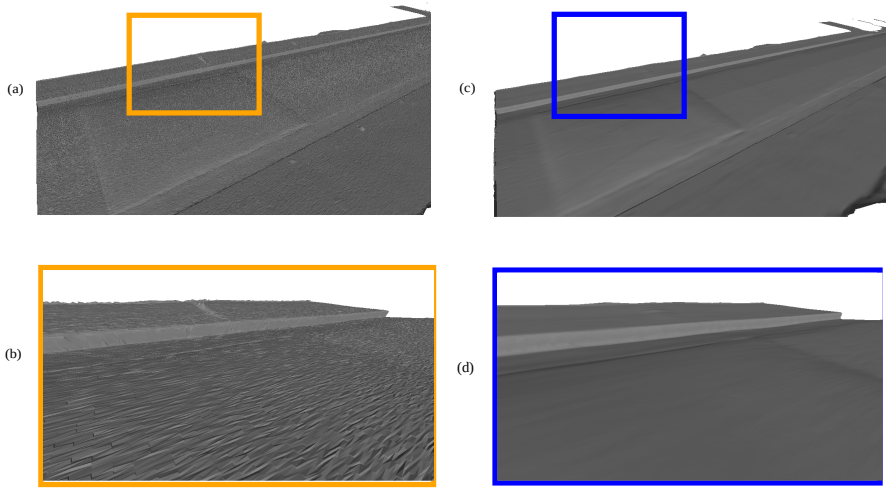


Fig. 4 The result of the Sinc windowed smoothing procedure [26] obtained on dataset Cassette acquired over the Cassette road situated in Paris, France. (a) the output of the Delaunay triangulation procedure, (b) zoom-in view on the area selected in the rectangle illustrated in Figure (a), (c) smoothed mesh output, (d) zoom-in view on the area selected in the blue rectangle illustrated in Figure (c).

7.4 Scalability

The smoothed mesh has a high number of triangles, being redundant and causing 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 [27], [28], mainly the default implementation available in the VTK library [31]. The mesh resolution $r(M_D)$ is controlled by the reduction factor, noted f_D.

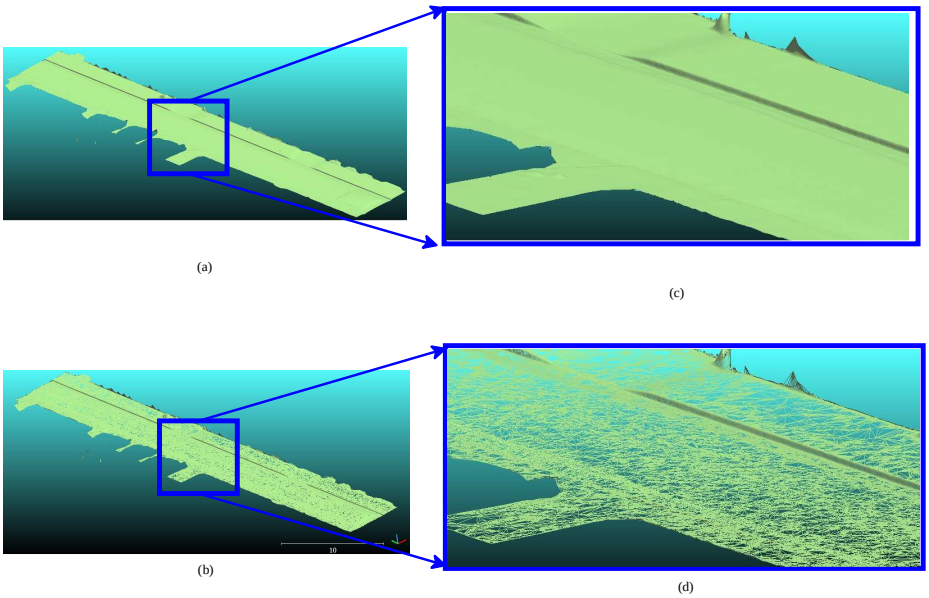


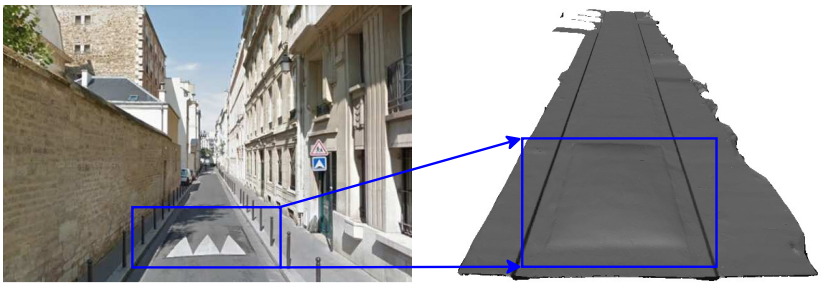
Fig. 5 The decimation results obtained for the dataset Assas: (a) smoothed mesh: $N_t = 2.54$ Mpts, (b) zoom-in view in of the area selected in the blue rectangle depicted in Figure (a), (c) decimated mesh, wire-frame view: decimation factor $f_D = 90\%$, N_t^D : 254 kTriangles, (d) zoom-in view of the area selected by the blue rectangle illustrated in Figure(c).

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 [27], following the vertex type (simple, interior, boundary, etc.), a different distance criterion is computed (distance to plane, distance to edge). Let us note with M_D the decimated mesh, and with N^D the corresponding number of triangles.

Figure 5 illustrates the result obtained for the input point cloud depicted in Figure 3 (c) reducing $f_D = 90\%$ of the entire mesh. The remaining number of triangles corresponds to a mesh resolution of $r(M_D) = 10\%$ 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 6 illustrate the speed bump reconstruction after applying a maximal mesh reduction factor of $f_D = 90\%$.

Accuracy of the decimated mesh. As in [32] we evaluate the accuracy of the decimated mesh by measuring the distance between the original point cloud $\mathbf{P}$ and the vertices of the decimated mesh, M_D. We choose to compute the Hausdorff distance [33], we study both, the mean and the root mean squared (RMS_H) distance for different mesh resolutions $r(M_D)$. 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.



(a) (b)

Fig. 6 The final output of the proposed surface reconstruction method obtained on the dataset Cassette, with 51 m length. (a) Google Maps view of the surveyed area, (b) $N_p = 1.01$ Mpts, $N_t = 2.026$ MTriangles, decimated mesh with $f_D = 90\%$, $N_t^D = 203$ kTriangles.

Table 1 Accuracy evaluation of the ground surface reconstruction with respect to ground truth (GT) data for Urban #2, (acquired over Cassette road situated in Paris, France) illustrated in Figure 6.

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)

8 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 Cassette (Urban #2) illustrated in Figure 6. It can be observed the reachable accuracy is better than 1.5 cm.

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 [34] and VTK [31] 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 representation 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 $r(M_S) = 10\%$. These results show that the proposed surface reconstruction framework provides a memory efficient surface representation, while preserving geometric details.

Table 2. Computation time for dataset Urban #2 illustrated in Figure 6 where P_S denotes the point cloud segmentation phase for the ground extraction; each column gives the runtime corresponding to each step of the algorithm. The overall computation time is about 17s for N_p=1.01 Mpts and N_t=203 kTriangles.

Steps	P_S	M_{DT}	M_C	M_S	M_D
CPU (s)	2	2.14	0.18	3	9

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 v_{rate} and measured using Cloud Compare [35] 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.

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 6.

Urban #2	**P**	M_{DT}	M_S	M_D
Memory (Mb)	14.856	81.61	37.600	3.7
v_{rate} (FPS)	267.74	10.273	12.448	131.96

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 frame- work, entitled Automatic Ground Surface Reconstruction (AGSR), with two well known surface reconstruction techniques. The first method is based on implicit functions [19], while the second is an explicit method [20] which proceeds by a greedy projection. Figure 7 and Table 4 illustrate the results obtained by applying each reconstruction algorithm to the point cloud **P** corresponding to the ground depicted in Figure 3 (c), acquired over Assas road (Paris, France), noted dataset Urban #1.

Table 4. Comparison between surface reconstruction methods: results obtained by running the algorithms on the dataset Urban #1 illustrated in Figure 3 (c): N_p^{out} and N_t^{out} denote the number of vertices and the number of triangles corresponding to each output, respectively.

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
	133	422	26

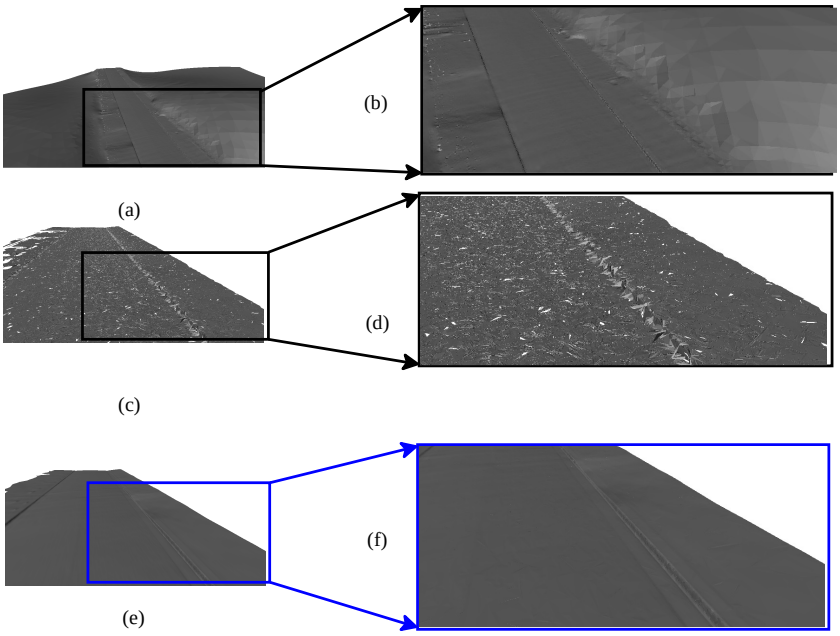


Fig. 7 Comparison of surface reconstruction results obtained for the dataset Assas (Urban #1): (a) Poisson technique, (b) zoom-in view of the rectangular area selected in Figure(a), (c) Greedy projection technique, (d) zoom-in view of the rectangular area selected in Figure (c), (e) proposed AGSR technique, (f) zoom-in view of the rectangular area selected in Figure (e).

By visually inspecting Figures 7 (a) and 7 (b), 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 7, but also in terms of computation time. More precisely, it allows to decrease the runtime5 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.

9 Scaling-Up Detail-Preserved 3D Ground Meshes

The proposed surface reconstructed algorithm was tested on several datasets acquired by two different MMS: Stereopolis [2 9] and L3D2 [30] equipped with Riegl and Velodyne sensing devices, respectively. Following the 3D sensor device, different smoothing parameters were used.

Two examples of the surface reconstruction result obtained for several datasets are depicted in Figure 8. For the dataset Cassette, the computation time for one chunck with 3 Mpts acquired along 50 m, the algorithm performs the surface reconstruction in about 17 s. For 100 chunks, the algorithm processes 100 Mpts representing the ground in about 28 min. When applied to long distance surveys, for 100 km non-stop driving and data-acquisition, the MMS acquires 6 Billion points and the ground surface reconstruction can be computed in about 10h. In this research work, we focus mainly on providing an accurate and scalable surface reconstruction algorithm, time scalability being beyond the scope of the paper. Nevertheless, an upgrade of computational resources by a factor of 10, results in real-time surface reconstruction capabilities. When such an upgrading scheme is adopted, the algorithm can deliver the entire road network for 10000 km length in about 5 days, non-stop driving, data acquisition and processing at 90 km/h.

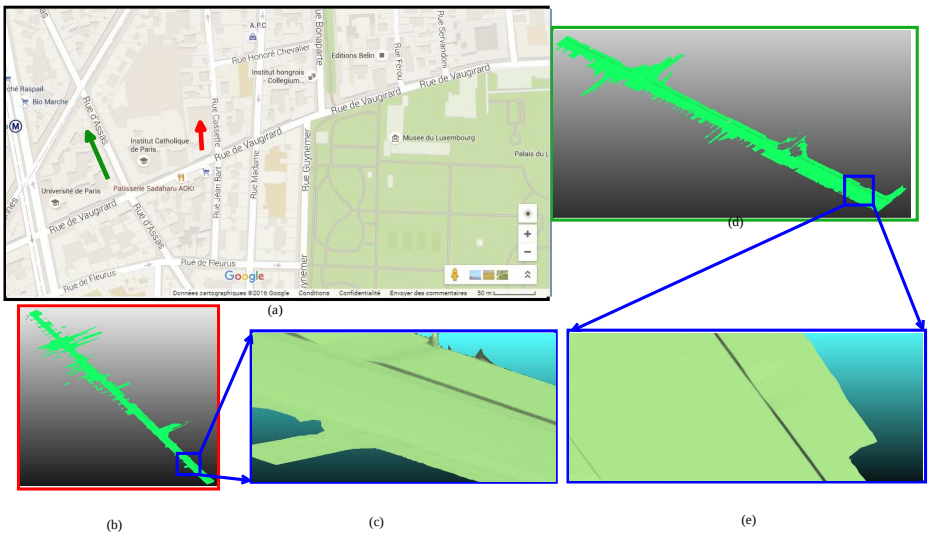


Fig. 8 Surface reconstruction results obtained for several datasets. (a) Google Maps view of two areas surveyed by the Sterepolis MMS: Assas road (Paris, France) - indicated by the green arrow, Cassette road (Paris, France) - indicated by the red arrow, (b) surface reconstruction results obtained for 5 scans segments, overall length: 319 m, (c) zoom-in view of the blue rectangular area presented in Figure (b), illustrating the accuracy of the reconstructed ramp access and sidewalks, (d) surface reconstruction results obtained for 4 chunks, overall length: 217 m, (e) zoom-in view of the blue rectangular area presented in Figure (d), illustrating the accuracy of the reconstructed ramp access and sidewalks.

10 Conclusions and Research Perspectives

The present research work introduces the Automatic Ground Surface Reconstruction algorithm designed to supply scalable and detail-preserving ground surface reconstruction in a fully automatic fashion. The proposed technique generates accurate 3D models of outdoor environments adapted for driving simulator engine

or for being embedded onboard mobile plateforms for autonomous navigation applications.

The reported technique addresses several open issues of the currently existing surface reconstruction techniques, such as: accurate reconstruction of sharp depth features in presence of noisy datasets, scalability and memory usage.

Research perspectives of the present research work are focusing on the photorealist surface reconstruction problem through the jointly use of laser reflectance and RGB cameras. A second research perspective is related to the facade surface reconstruction and ground-faade merging within a global referential frame.

Daniela Craciun is a Research Scientist in Applied Mathematics to Computer Vision and Robotics fields. She achieved her Ph. D. degree at Telecom ParisTech ENST along with a doctoral grant funded by the French Mapping Agency to work on 3D modelling of complex environments from active and passive 3D vision sensors. In 2013, she joined the Robotics Center of the MINES ParisTech to work on 3D modelling from mobile mapping systems. She is currently with the CNAM – iMATH Department conducting research on 3D shape modelling and pattern recognition.

Jean-Emmanuel Deschaud is now an assistant professor at the Robotics Center in MINES ParisTech school, Paris, France. After a PhD in Computer Science in 2010 from MINES ParisTech, he was a Post-Doc for one year at the Robotics Institute of Carnegie Mellon University in Pittsburgh, USA. He is now working on 3D Computer Vision, more precisely on processings of lidar data (segmentation, classification, reconstruction and rendering).

François Goulette is a full Professor at MINES ParisTech. He also graduated himself from MINES ParisTech (MSc, PhD) and holds an Habilitation à Diriger des Recherches (HDR) from University Pierre and Marie Curie. He has worked in robotics research for Electricité de France (EDF), and has been a Visiting Scientist at the Artificial Intelligence Laboratory. He is leading a group on 3D Computer Vision, related to Robotics, Photogrammetry and Remote Sensing. He has been recently active on Laser-based Mobile Mapping Systems acquisition, rendering and automatic data processing.

References

[1] Despine, G., and C. Baillard. Realistic road modelling for driving simulators using GIS data. *In Advances in Cartography and GIScience, Lecture Notes in Geoinformation and Cartography*, 2011.

[2] SIMVIR (web page). Available from: http://crvm.ism.univ-amu.fr/simvir/index.html

[3]Bouchner, P., and S. Novotny. Car dynamics model design for interactive driving simulation use. *In Recent Researches in Applied Informatics, 2nd Proceeding of Int. Conference on Applied Mathematics and Computing Theory*, 2011.

[4] Petit, J. Génération, visualization et évaluation d'images (HDR). Application la simulation de conduite nocturne. *PhD Dissertation, Université Claude Bernard Lyon I, 2010.*

[5] Brémond, R. La visibilite routière: une approche pluri-disciplinaire. *HdR Université Paris-Est, 2010.*

[6] Chaperon, T., and J. P. Colinot. The new PSA Peugeot-Citroen advanced driving simulator: Overall design and motion cue algorithm. *In Driving Simulation Conference - North America*, 2007.

[7] A. Bolling J. Jansson, A. Genell M. Hjort M. Lidstrm S. Nordmark G. Palmqvist H. Sehammar L. Sjgren M. Ogren. Shake: an approach for realistic simulation of rough roads in a moving base driving simulator. *In Driving Simulation Conference - Europe 2010.*

[8] CityGML (web page). Available from: http://www.citygml.org/

[9] RoadXML (web page). Available from: http://www.road-xml.org/

[10] Oktal (webpage). Available from: http://www.oktal.fr/fr/

[11] Carlberg, M., J. Andrews, P. Gao, and A. Zakhor. Fast surface reconstruction and segmentation with ground-based and airborne lidar range data. In *International Conference 3D PVT*, 2008.

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

[13] Yu, S. J., S. R. Sukumar, A. F. Koschan, D. L. Page, and M. A. Abidi. 3D reconstruction of road surfaces using an integrated multi-sensory approach. *In Optic and Lasers in Enginnering*, 45:808–818, 2007.

[14] Jaakkola, A., J. Hyyppa, H. Hyyppa, and A. Kukko. Retrevial algorithms for road surface modelling using laser-based mobile mapping. *In Sensors, Open Access, 2008.*

[15] Alexa, M., J. Behr, D. Cohen-or, S. Fleishman, D. Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Computer Visualization and Graphics* 9: 3–15, 2003.

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

[17] Edelsbrunner, Herbert. Shape reconstruction with Delaunay complex. *Proceedings of the Third Latin American Symposium on Theoretical Informatics, Springer-Verlag* 1: 119–132, 1998.

[18] Gopi, M., S. Krishnan, and C. T. Silva. Surface reconstruction based on lower dimensional localized Delaunay triangulation. In *Computer Graphics Forum* 19:467-478, 2000.
[19] Kazhdan, M., M. Bolitho, and H. Hoppe. Poisson surface reconstruction. *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, 2006.
[20] Marton, Z. C., R. B. Rusu, and M. Beetz. On fast surface reconstruction methods for large and noisy point clouds. *In IEEE International Conference on Robotics and Automation*, 2009.
[21] Bohren, Jonathan, Tully Foote, Jim Keller, Alex Kushleyev, Daniel Lee, Alex Stewart, Paul Vernaza, Jason Derenick, John Spletzer, and Brian Satterfield. Litte ben: The ben franklin racing team's entry in the 2007 DARPA urban challenge. *Journal of Field Robotics, Special Issue on the 2007 DARPA Urban Challenge* 25: 598–614, 2008.
[22] Serna, A., and B. Marcotegui. Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. *In Press* 28: 493–501, 2014.
[23] Matheron, G. Random sets and integral geometry. *John Wiley & Sons, New York* 28: 493–501, 1975.
[24] Serra, J. Image analysis and mathematical morphology: Theoretical advance. *Academic Press* 2: 493–501, 1988.
[25] Shewchuk, Jonathan Richard. 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: 203–222, 1996.
[26] Taubin, G., T. Zhang, and G. Golub. Optimal surface smoothing as filter design. *IBM Research Report* 1148: 203–222, 1996.
[27] Schroeder, W. J., J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *In Conference Proceedings of SIGGRAPH*, 1992.
[28] Hoppe, H. Progressive meshes. *In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, 1996.
[29] Paparoditis, N., J. Papelard, B. Cannelle, A. Devaux, B. Soheilian, N. David, and E. Houzay. 2012. Stereopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualization and 3D metrology. *Revue Francaise de Photogrammétrie et de Télédétection*, 2012.
[30] Goulette, F., F. Nashashibi, I. Abuhadrous, S. Ammoun, and C. Laurgeau. 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*, 2006.
[31] Schroeder, W., K. Martin, and B. Lorensen. The visualization toolkit. *Third Edition Kitware Inc.*, 2006.
[32] Turnet, E., and Avideh Zakhor. Watertight planar surface meshing of indoor points clouds with voxel carving. *In IEEE International Conference on 3D Vision*, 2013.
[33] Cignoni, Paolo, Claudio Rocchini, and Roberto Scopigno. 1998. Measuring error on simplified surfaces. *Computer Graphics Forum* 17: 167–174.
[34] Rusu, Radu Bogdan, and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
[35] Cloud Compare (web site). Available from: *http://www.danielgm.net/cc/*.