SCALABLE AND DETAIL-PRESERVING GROUND SURFACE RECONSTRUCTION FROM MOBILE LASER SYSTEMS FOR DRIVING SIMULATORS ENGINES

Daniela Craciun, Jean-Emmanuel Deschaud, François Goulette

SCALABLE AND DETAIL-PRESERVING GROUND SURFACE RECONSTRUCTION FROM MOBILE LASER SYSTEMS FOR DRIVING SIMULATORS ENGINES

Daniela Craciun[1], Jean-Emmanuel Deschaud[1] and François Goulette[1]

(1) : Ecole des Mines de Paris, Centre de Robotique, 60 Bd. Saint Michel, 75272 Paris Cedex 6
0033(0)1 40 51 94 54, Fax 0033 (0)1 43 26 10 51
E-mail : {daniela.craciun,jean-emmanuel.deschaud, francois.goulette}@mines-paristech.fr

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 driver's training. This paper presents the **A**utomatic **G**round **S**urface **R**econstruction (**AGSR**) method, a framework which exploits 3D data acquired by mobile laser 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.

Key words: surface reconstruction, LiDAR, driving simulator engines, road network, mobile laser systems.

1. Introduction and Motivation

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 [Des7]. However, the resulted 3D models lack geometrical accuracy and photorealism, limiting therefore driver's 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 driver's perception for car's maneuverability.

In order to overcome the limitations of the existing 3D road modeling methods, several research projects [SIM21] are directed towards the use of Mobile Laser Systems (MLS) which allow sensing the environment of their surroundings with high sampling rates at high vehicle velocities. Such 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. 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 modelled correspondingly. In addition, in structured environments such as urban scenes, sharp depth features and geometrical details must be preserved. This is a key aspect for driving simulator engines which require a very accurate surface modeling of road borders, accessibility ramps and other geometric details. Smoothing the noise in MLS data while preserving sharpness is still a difficult task for the existing surface reconstruction algorithms. Furthermore, in order to perform surface reconstruction automatically over large distances, memory constraints and scalability issues must be addressed.

The research work reported in this paper aims at exploiting 3D data acquired by a MLS for generating automatically geometrically accurate surface reconstruction of 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, while fulfilling the requirements of driving simulation engines. The paper is organized as following. Section 2 introduces our method for improving perceptive realism from MLS data and the implementation of ground 3D models within the simulator software. The next section presents the overview of the proposed ground surface framework. Sections 4 and 5 are dedicated to a description of the two main phases of our ground surface reconstruction algorithm. Sections 6 and 7 present the performance evaluation, including scalability, followed by Section 8 which resumes the proposed method.

2. Perceptive Realism from MLS Data

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 developing 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 [Bou1]. The spatio-temporal coherence in a driving simulation engine is a major concern. It is related to the proprioceptive integration, i. e.: human's sensibility to delay and perception incoherence (depth, motion) [Pet15, Bre2]. 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 [Cha4], 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, car's vibrations [Bol3, Seh19] and sound component.

Visual layer from MLS data. The use of GIS 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 using 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, [SIM21] 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 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 [Cit6] or RoadXML [Roa16], accepted by driving simulation engines. They are usually widely employed to supply the software of driving simulation engines. A good example is SCANeR™ [Okt13] which provides a complete description of roads network for a variety of driving simulation engines.

3. Automatic Ground Surface Reconstruction (AGSR)

The overall processing workflow of the proposed surface reconstruction method comes together with the global framework illustrated

in Fig. 1 which has as input a massive 3D point cloud acquired by a MLS. 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. Fig. 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. 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.

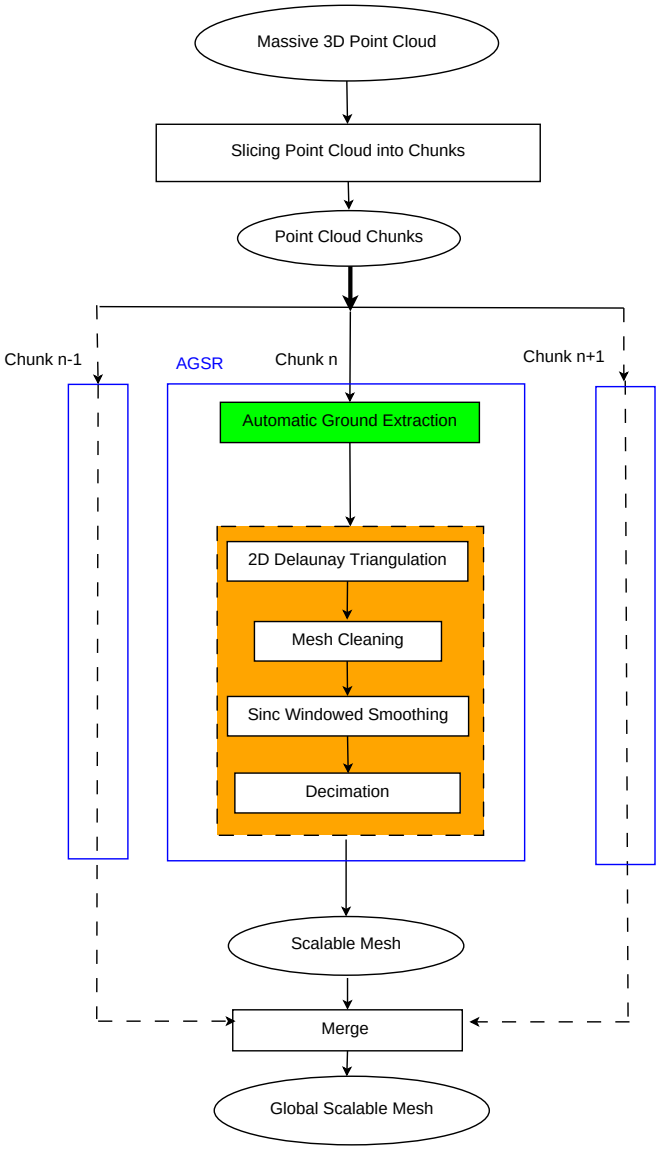

Fig. 1. The workflow of the proposed Automatic Ground Surface Reconstruction (AGSR) framework designed to be integrated within a global parallel computation scheme dedicated to massive 3D point cloud processing.

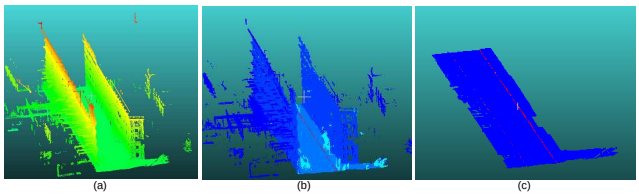

Fig. 2. Ground extraction results: (a) example of a 3D chunk acquired over Assas street located in Paris (France): approximative length of 82 m, N=3 Mpts. For visualization purposes, the original point cloud is colored with respect to elevation values; (b) 3D object segmentation results; façade – dark blue, road and sidewalks – blue, non-ground objects – light blue, sidewalk borders and ramp access – red; (c) the extracted ground corresponding to road, sidewalk and ramp access: 1.27 Mpts.

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 [Ser19] which assigns semantic labels with respect to different classes: ground, 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.

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 and decimation techniques to generate automatically a regular and scalable mesh representation of the ground. Each decimated mesh resulted from the surface reconstruction method is further merged within a global reference scene. Note that the entire workflow can be applied in parallel to each chunk, while merging each current ground surface with a global scene, on the fly, as they are computed.

This paper is concerned with the automatic ground surface reconstruction (AGSR) phases, mainly its extraction and surface reconstruction procedures which are described in the following two sections.

4. 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 [Her9, Ser19]. It is based on elevation images and it

relies on image processing techniques, especially Mathematical Morphology [Mat11, Mer12]. The general workflow is composed by several steps: first, the 3D point cloud is projected to an elevation image. 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 procedure $Fill(f)$ is used since this transformation does not create new regional maxima in the 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. This section resumes the ground segmentation and classification steps. For further details and complete analysis in each step, the reader is encouraged to review [Ser19].

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 [Her9]. It is based on the λ-flat zones labeling algorithm defined in [Mey12]. The parameter λ 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. Fig.2 (b) presents the segmentation and classification result corresponding to the input point cloud illustrated in Fig. 2 (a). Fig.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 Fig. 2 (c)), into a continuous and scalable surface representation. The proposed framework is composed by several steps which are illustrated in Fig. 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 [Tau22] smoothing algorithm which eliminates high frequencies, while preserving sharp depth features and avoiding surface shrinkage. In a final step, a progressive decimator [Hop10] is applied to the smoothed mesh in order to cope with scalability constraints when performing surface reconstruction over large 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 $\mathbf{P} = \left\{ x_i, y_i, z_i \middle| i = 1,..., N_p \right\}$ the 3D point cloud corresponding to the ground, where N_p denotes the number of points. We apply the Triangle algorithm [She20] to the 3D point cloud **P** to generate a planar constraint Delaunay triangulation which provides the connectivity between points. Let us note with 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 $e_i^j \in M_{DT}$, i.e. over all triangles $t^j \in M_{DT}, j = 1,.., N_t$ and for its corresponding edges $e_i^j, i = \{1,2,3\}$. The term δ denotes a proportionality factor. A triangle t^j is eliminated if any of its edges $e_i^j > e_{\max}, i = \{1,2,3\}$. In practice, for several datasets acquired by different MLS systems, we found that a coefficient $\delta = 20$ results in a mesh without long triangles, which we note 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 [Tau22] 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 M_S; it can be observed that the Sinc Windowed smoothing technique provides a regular surface, while preserving roads and sidewalk borders sharpness.

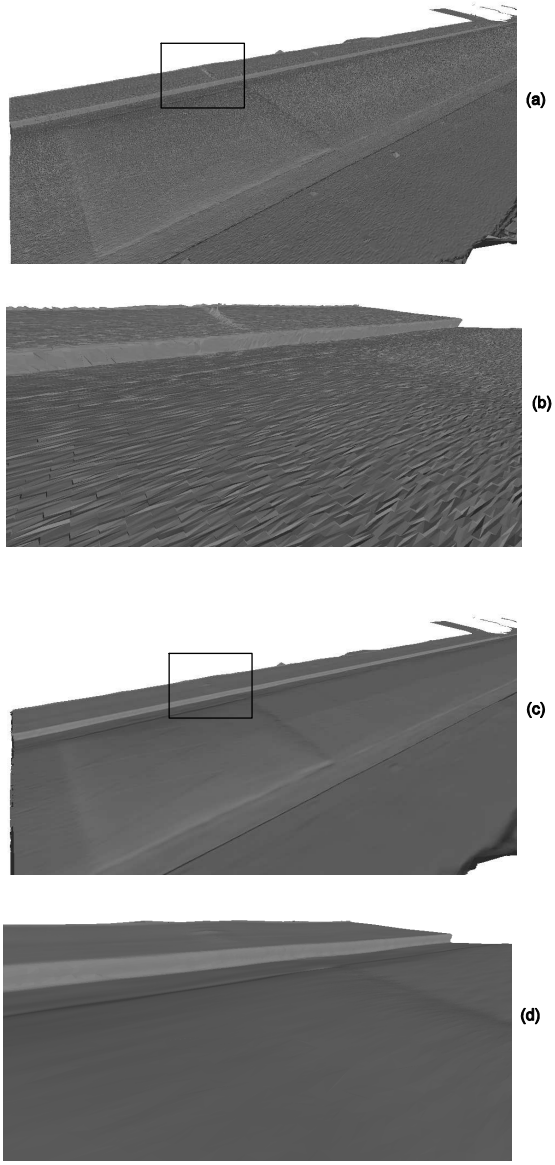

Fig. 3. Mesh smoothing results on dataset Cassette acquired over Cassette street situated in Paris, (France) by the STEREOPOLIS MMS [Pap14] equipped with a Riegl sensing device. (a) planar Delaunay triangulation, (b) zoom-in view on the sidewalk border corresponding to the black rectangle area illustrated in Fig. (a); (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 Fig. (c).

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 described in [Zar24, Hop10]. The resolution of the decimated mesh M_D given by $r(M_D)$ is controlled by the reduction factor, noted $f_D(\%)$. A second advantage of the progressive decimation algorithm is that it can generate progressive meshes in an incremental fashion for efficient visual rendering. The algorithm proceeds as follows: first, each vertex is classified and inserted in a priority queue for further processing. The priority is set following the error caused by the vertex elimination and by the re-triangulation of the resulting hole. Let us note with N_t^D the number of triangles of the decimated mesh. Fig. 4 illustrates the result obtained for the input depicted in Fig. 2 (c) reducing $f_D = 90\%$ of the entire mesh. The remaining number of triangles corresponds to $r(M_D) = 10\%$ 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, Fig. 5 illustrate the speed bump reconstruction after applying a maximal reduction factor of $f_D = 90\%$.

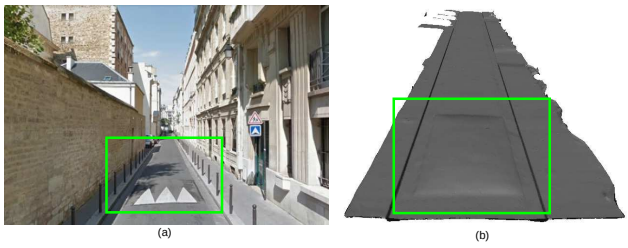

Fig. 4. Surface reconstruction results obtained for the dataset Cassette. (a) Google street view of the surveyed area; (b) Np = 1.01 Mpts, Nt=2.026 MTriangles, decimated mesh with fD = 90%, NDT = 203kTriangles.

Accuracy of the decimated mesh. As in [Tur23], we evaluate the accuracy of the decimated mesh by measuring the distance between the original point cloud **P** and the vertices of the decimated mesh, M_D. We choose to compute the Hausdorff distance [Cig5] and study both, the mean and the root mean squared distance RMS_H 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 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. In order to quantify the accuracy of the reconstructed surface, we perform several measurements on site, mainly: the height of the sidewalk border and the height of the access ramp, noted $H_{sidewalk}$ and H_{ramp}, respectively. Table 1 illustrates the ground truth and the reconstructed dimensions for dataset Cassette shown in Fig. 4. It can be observed the reachable accuracy is better than 1.5 cm.

Table 1. Accuracy evaluation of the ground surface reconstruction algorithm with respect to ground truth (GT) data for dataset Cassettte.

Measurements	$H_{sidewalk}$ (cm)	H_{ramp} (cm)
GT	10.5	2.5
Reconstruction	10.1	2.3

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 [Rus17] and VTK [Sch18] libraries. Table 2 illustrates the computation time obtained for the dataset Cassette. 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.

Table 2: Computation time for dataset Cassette illustrated in Fig. 4 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 17 s for N_p = 1.01 Mpts, N_t=203 kTriangles.

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

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.

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 υ_{rate} and measured using Cloud Compare [Gir8] 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. 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. It can be observed that even though 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.

Table 3: Memory usage and frame frequency measures corresponding to the input chunk P and and to the main outputs of the algorithm for dataset Cassette illustrated in Fig. 4.

Cassette	P_S	M_{DT}	M_S	M_D
Memory (Mb)	14.85	81.61	37.6	3.7
υ_{rate} (fps)	267.74	10.27	12.44	131.96

7. Scaling-Up Cartography

We test the surface reconstruction algorithm on several chunks from several urban datasets acquired with different MLS systems. Fig. 5 illustrates the results obtained on 4 scans segments acquired by STEREPOLIS mobile platform [Pap14] using a Riegl sensor. The entire scene contains 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 Cassette, in average, we process 3 Mpts for 50 m length of surveyed area in about 17 s.

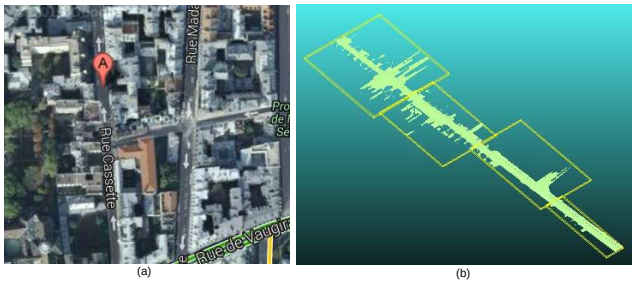

Fig. 5. Surface reconstruction results obtained for dataset Cassette. (a) Google street view of the surveyed area, (b) surface reconstruction results obtained for 4 chunks with different lengths (each one included in its bounding box), overall approximative distance: 217 m.

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 could 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 the **A**utomatic **G**round **S**urface **R**econstruction (**AGSR**), a fully automatic framework for generating a scalable and detail-preserving ground surface reconstruction from MLS data in outdoor environments. The presented method addresses several key issues of the currently existing surface reconstruction methods such as: accurate reconstruction of sharp depth features in presence of noisy data sets, scalability and memory usage for efficient data transmission and visualization over large distances.

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 distribute the computation of different surface reconstruction frameworks. Thanks to the segmentation and classification algorithm, the entire framework can be combined with geometrically consistent surface reconstruction framework in order to expand the 3D modeling capability to buildings and non-ground objects (trees, cars, urban furniture). When such a multiple target reconstruction scheme is employed, its parallelization is straightforward. The proposed framework emphasizes the high potential of the MLS 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 photorealist surface reconstruction problem through the jointly use of laser reflectance and RGB cameras. Research perspectives are also concerned with the extension of the global workflow to facade reconstruction, while tacking into account ground and façade merging within a global referential frame.

References

[Bou1] P. Bouchner and S. Novotný. 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.

[Bre2] R. Brémond. La visibilitié routière: une approche pluri-disciplinaire. HdR, 2010.

[Bol3] A. Bolling, J. Jansson, A. Genell, M. Hjort, M. Lidström, S. Nordmark, G. Palmqvist, H. Sehammar, L. Sjögren, M. Ogren. SHAKE – an approach for realistic simulation of rough roads in a moving base driving simulator. In Driving Simulation Conference - Europe, 2010.

[Cha4] T. Chaperon and J.-P. Colinot. The new PSA Peugeot-Citroën Advanced Driving Simulator: Overall design and motion cue algorithm. In Driving Simulation Conference - North America, 2007.

[Cig5] P. Cignoni, C. Rocchini. and R. Scopigno. Measuring error on simplified surfaces. In Conference on Computer Graphics Forum, pp. 167–174, 1998.

[Cit6] CityGML: http://www.opengeospatial.org/standards/citygml

[Des7] G. Despine and C. Baillard. Realistic Road Modelling for Driving Simulators using GIS Data. In Advances in Cartography and GIScience, Lecture Notes in Geoinformation and Cartography, pp 431-448, 2011.

[Gir8] D. Girardeau-Montaut. Telecom Paristech & EDF. Cloud Compare, site web: http://www.danielgm.net/cc/, 2014.

[Her9] J. Hernández, and B. Marcotegui. Filtering of artifacts and pavement segmentation from mobile lidar data. In ISPRS workshop Laser scanning, XXXVIII-3/W8 2, pp. 329–333, 2009.

[Hop10] H. Hoppe. Progressive meshes. In Proceedings of 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH pp. 99–108, 1996.

[Mat11] G. Matheron. Random sets and integral geometry. John Wiley & Sons, New York 28, pp. 493–501, 1975.

[Mey12] F. Meyer, From connected operators to levelings. 12, pp. 191–198, 1998.

[Okt13] Oktal site web: www.oktal.fr, 2014.

[Pap14] N. Paparoditis, J.-P. Papelard, B. Cannelle, A. Devaux, B. Soheilian, N. David and E. Houzay. 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, Vol. 200, pp. 69-79, 2012.

[Pet15] J. Petit. Génération, visualisation et évaluation d'images HDR. Application à la simulation de conduite nocturne. PhD Dissertation, 2010.

[Roa16] RoadXML site web:

http://en.wikipedia.org/wiki/RoadXML, 2014.

[Rus17] R. B. Rusu and S. Cousins. 3D is here: Point cloud library (pcl). IEEE International Conference on Robotics and Automation (ICRA), 2011.

[Sch18] W. Schroeder, K. Martin and B. Lorensen. The visualization toolkit. Third Edition Kitware Inc, 2006.

[Ser19] A. Serna and B. Marcotegui. Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning. In Press 28, pp. 493–501, 2014.

[She20] J. R. Shewchuk. 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, 1996.

[SIM21] SIMVIR site web: http://www.realite-virtuelle.univmed.fr/simvir/index.html, 2014.

[Tau22] G. Taubin and G. Golub. Optimal surface smoothing as filter design. IBM Research Report 1148, pp. 203–222, 1996.

[Tur23] E. Turnet and A. Zakhor. Watertight planar surface meshing of indoor points-clouds with voxel carving. In IEEE International Conference on 3D Vision pp. 41–48, 2013.

[Zar24] W. J. Schroeder. J. A. Zarge and W. E Lorensen. Decimation of triangle meshes. In Conference Proceedings of SIGGRAPH, pp. 65–70, 1992.