On the Production of Semantic and Textured 3D Meshes of Large scale Urban Environments from Mobile Mapping Images and LiDAR scans

Mohamed Boussaha, Eduardo Fernandez-Moral, Bruno Vallet, Patrick Rives

On the Production of Semantic and Textured 3D Meshes of Large scale Urban Environments from Mobile Mapping Images and LiDAR scans

Mohamed Boussaha[1] Eduardo Fernandez-Moral[2] Bruno Vallet[1] Patrick Rives[2]

[1] IGN/LASTIG MATIS, Université Paris Est, 73 avenue de Paris 94160 Saint-Mandé, France
[2] Inria Sophia Antipolis-Méditerrané, Lagadic, 2004 route de Lucioles- BP93, 06902 Sophia Antipolis, France

(mohamed.boussaha, bruno.vallet)@ign.fr
(eduardo.fernandez-moral, patrick.rives)@inria.fr

Résumé

Dans cet article nous présentons un cadre entièrement automatique pour la reconstruction d'un maillage, sa texturation et sa sémantisation à large échelle à partir de scans LiDAR et d'images orientées de scènes urbaines collectés par une plateforme de cartographie mobile terrestre. Tout d'abord, les points et les images georéferencés sont découpés temporellement pour assurer une cohèrence entre la geométrie (les points) et la photométrie (les images). Ensuite, une reconstruction de surface 3D simple et rapide basée sur la topologie d'acquisition du capteur est effectuée sur chaque segment après un rééchantillonnage du nuage de points obtenu à partir des balayages LiDAR. L'algorithme de [31] est par la suite adapté pour texturer la surface reconstruite avec les images acquises simultanément assurant une texture de haute qualité et un ajustement photométrique global. Enfin, en se basant sur le schéma de texturation, une sémantisation par texel est appliquée sur le modèle final.

Mots Clef

scène urbaine, cartographie mobile, LiDAR, reconstruction de surface, texturation, sémantisation, apprentissage profond.

Abstract

In this paper we present a fully automatic framework for the reconstruction of a 3D mesh, its texture mapping and its semantization using oriented images and LiDAR scans acquired in a large urban area by a terrestrial Mobile Mapping System (MMS). First, the acquired points and images are sliced into temporal chunks ensuring a reasonable size and time consistency between geometry (points) and photometry (images). Then, a simple and fast 3D surface reconstruction relying on the sensor space topology is performed on each chunk after an isotropic sampling of the point cloud obtained from the raw LiDAR scans. The method of [31] is subsequently adapted to texture the reconstructed surface with the images acquired simultaneously, ensuring a high quality texture and global color adjustment. Finally, based on the texturing scheme a per-texel semantization is conducted on the final model.

Keywords

urban scene, mobile mapping, LiDAR, surface reconstruction, texturing, semantization, deep learning.

1 Introduction

Representing and understanding the 3D geometric and photometric information of the real world using mobile mapping data is one of the most challenging and extensively studied research topics in the photogrammetry and robotics communities. Such particular focus can be explained by the increasing trend of using hybrid mobile mapping systems acquiring both images and LiDAR point clouds of the environment. However, these two modalities remain basically exploited independently for multiple tasks(segmentation, classification, localization ...) while a joint exploitation of these sources of information would benefit not only from their complementarity (accurate point clouds of the LiDAR vs high resolution of images), but also from their different acquisition geometries. In this paper we propose a fusion of image and LiDAR data into a single representation : a textured semantic 3D mesh. We believe that such representation will be a core component of several real world applications in urban planning and modeling, city navigation and autonomous platforms technologies. Our contributions are as follows :

- proposing a simple reconstruction approach based on the sensor space topology.
- adapting the state of the art texturing method [31] to mobile mapping images and LiDAR scans.
- taking advantage of the texturing scheme to assign a label to each texel by projecting the semantic predictions of spherical images (constructed from the acquired oriented images and labeled using the method of [1]) onto the 3D mesh.

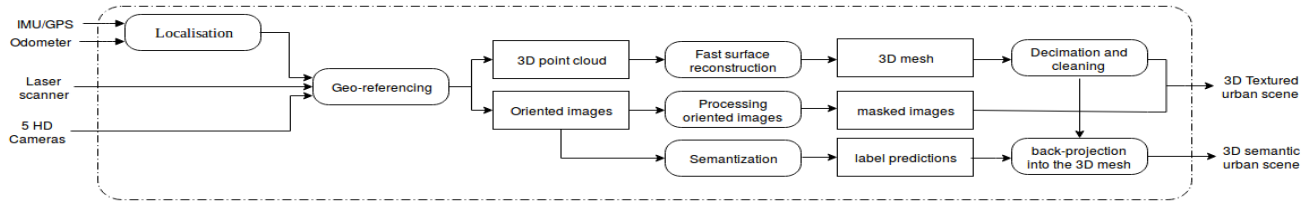


FIGURE 1 – The pipeline for producing textured and semantized 3D meshes

1.1 Related work

In the following we give an overview of the various methods related to the design of our pipeline.

Texture mapping : from the robotics community perspective, conventional 3D urban mapping approaches usually propose to use LiDAR or camera separately but a minority has recently exploited both data sources to build dense textured maps [27]. In the literature, both image-based methods [32, 17, 28] and LiDAR-based methods [11, 13] often represent the map as a point cloud or a mesh relying only on geometric properties of the scene and discarding interesting photometric cues while a faithful 3D textured mesh representation would be useful for not only navigation and localization but also for photo-realistic accurate modeling and visualization.
The computer vision and the computer graphics communities have generated compelling urban texturing results. [29] developed an interactive system to texture architectural scenes with planar surfaces from an unordered collection of photographs based on structure-from-motion. [8] perform impressive work by texturing entire cities. Still, they are restricted to 2.5D scene representation and they also operate exclusively on regular block city structures with planar surfaces and treat buildings, ground, and building-ground transitions differently. In order to achieve a consistent texture across patch borders in a setting of unordered registered views, [4, 10] choose to blend these multiple views by computing a weighted cost indicating the suitability of input image pixels for texturing with respect to angle, proximity to the model and the proximity to the depth discontinuities. However, blending images induces strongly visible seams in the final model especially in the case of a multi-view stereo setting because of the potential inaccuracy in the reconstructed geometry.

While there exists a prominent work on texturing urban scenes, we argue that **large scale** texture mapping should be **fully automatic** without the user intervention and efficient enough to handle its computational overhead in a reasonable time frame. In contrast to the latter methods, [31] proposed to select a single view per face based on a pairwise Markov random field taking into account the viewing angle, the proximity to the model and the resolution of the image. Then, color discontinuities are properly adjusted by looking up the vertex' color along all adjacent seam edges. We consider [31] as a base for our work since it is the first comprehensive framework for texture mapping that enables fast and scalable processing.
3D semantic segmentation : since the huge success of deep learning techniques in 2D semantic segmentation, multiple attempts to extend these approaches to a 3D setting especially point clouds have been proposed [23, 24, 25, 6]. However, these methods are limited by numerous challenges, the most obvious one being the scale of the data making the use of direct deep learning on raw point clouds intractable. In order to overcome this problem, other interesting alternatives have been presented. For example SnapNet [2] proposed to generate a set of 2D virtual views (RGBD images) from the underlying mesh, the semantic labeling of which is subsequently projected onto the 3D model. SEGCloud [30] handles large clouds by subsampling and then uses 3D convolutions on a regular voxel grid. [26] first generates mesh models using multi-view reconstruction, then a Conditional Random Field (CRF) based on hand-crafted geometric and photometric features is proposed to efficiently determine which image is best suited to capture the semantic assignment of the face obviating the need to label the redundant (overlapped parts of the images) set of views, an inherited problem of the multi-view reconstruction setting.
In our work, we abstain from the multi-view surface reconstruction step for multiple reasons. As pointed out above, methods based on structure-from-motion and multi-view stereo techniques usually yield to less accurate camera parameters, hence the reconstructed geometry might not be faithful to the underlying model compared to LiDAR based methods [22] which results in ghosting effect and strongly visible seams in the textured model. Besides, such methods do not allow a direct and automatic processing on raw data due to relative parameters tuning for each dataset and in certain cases their computational cost may become prohibitive. Instead, we propose a simple but fast algorithm to construct a mesh from the raw LiDAR scans.
In Figure 1, we depict the whole pipeline to generate large scale textured and semantic models leveraging on the georeferenced raw data. Then, we construct a 3D mesh representation of the urban scene and subsequently fuse it with the preprocessed images (masked and labeled images) to get a 3D textured and semantic city models.
The rest of the paper is organized as follows : In Section 2 we present a fast and scalable mesh reconstruction algo-

rithm. The semantization process is discussed in Section 3. Section 4 explains the texture and label predictions mapping approach. We show our experimental results in Section 5. Finally, in Section 6, we conclude the paper.

2 Sensor-topology based surface reconstruction

In this section, we propose an algorithm to extract a large scale mesh on-the-fly using the point cloud structured as series of line scans gathered from the LiDAR sensor being moved through space along an arbitrary path.

2.1 Mesh extraction process

During urban mapping, the mobile platform may stop for a moment because of external factors (e.g. road sign, red light, traffic congestion ...) which results in massive redundant data at the same scanned location. Thus, a filtering step is mandatory to filter out redundant scan lines. To do so, we fix a minimum distance between two successive line scans and we remove all lines whose distances to the previous (unremoved) line is less than a fixed threshold. In practice, we use a threshold of $1cm$, close to the LiDAR accuracy.

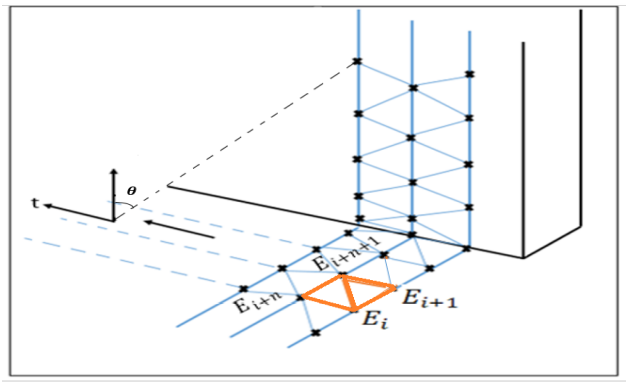


FIGURE 2 – Triangulation based on the sensor space topology

Once the regular sampling is done, we consider the resulting point cloud in the sensor space where one dimension is the acquisition time t and the other is the θ rotation angle. Let θ_i be the angle of the i^{th} pulse and E_i the corresponding echo. In case of multiple echoes, E_i is defined as the last (furthest) one, and in case of no return, E_i does not exist so we do not build any triangle based on it. In general, the number N_p of pulses for a 2π rotation is not an integer so E_i has six neighbors E_{i-1}, E_{i+1}, E_{i-n}, E_{i-n-1}, E_{i+n}, E_{i+n+1} where $n = \lfloor N_p \rfloor$ is the integer part of N_p. These six neighbors allow to build six triangles. In practice, we avoid creating the same triangle more than once by creating for each echo E_i the two triangles it forms with echoes of greater indices : E_i, E_{i+n}, E_{i+n+1} and E_i, E_{i+n+1}, E_{i+1} (if the three echoes exist) as illustrated in Figure 2. This allows the algorithm to incrementally and quickly build a triangulated surface based on the input points of the scans. In practice, the (non integer) number of pulses N_p emitted during a $360\,\mathrm{deg}$ rotation of the scanner may slightly vary, so to add robustness we check if $\theta_{i+n} < \theta_i < \theta_{i+n+1}$ and if it doesn't, increase or decrease n until it does.

2.2 Mesh cleaning

The triangulation of 3D measurements from a mobile mapping system usually comes with several imperfections such as elongated triangles, noisy unreferenced vertices, holes in the model, redundant triangles ...to mention a few. In this section, we focus on three main issues that frequently occur with mobile terrestrial systems and affect significantly the texturing results if not adequately dealt with.

Elongated triangles filtering. In practice, neighboring echoes in sensor topology might belong to different objects at different distances. This generates very elongated triangles connecting two objects (or an object and its background). Such elongated triangles might also occur when the MMS follows a sharp turn. We filter them out by applying a threshold on the maximum length of an edge before creating a triangle, experimentally set to $0.5m$ for the data used in this study.

Isolated pieces removal. In contrast with camera and eyes that captures light from external sources, the LiDAR scanner is an active sensor that emits light itself. This results in measurements that are dependent on the transparency of the scanned objects which cause a problem in the case of semitransparent faces such as windows and front glass. The laser beam will traverse these objects, creating isolated pieces behind them in the final mesh. To tackle this problem, isolated connected components composed by a limited number of triangles and whose diameter is smaller than a user-defined threshold are automatically deleted from the final model.

Hole filling. After the surface reconstruction process, the resulting mesh may still contain a consequent number of holes due to specular surfaces deflecting the LiDAR beam, occlusions and the non-uniform motion of the acquisition vehicle. To overcome this problem we use the method of [14]. The algorithm takes a user-defined parameter which consists of the maximum hole size in terms of number of edges and close the hole in a recursive fashion by splitting it until it gets a hole composed exactly with 3 edges and fills it with the corresponding triangle.

2.3 Scalability

The interest in mobile mapping techniques has been increasing over the past decade as it allows the collection of dense and very accurate and detailed data at the scale of an entire city with a high productivity. However, processing such data is limited by various difficulties specific to this type of acquisition especially the very high data volume which requires very efficient processing tools in terms of number of operations and memory footprint. In order to perform an

automatic surface reconstruction over large distances, memory constraints and scalability issues must be addressed. First, the raw LiDAR scans are sliced into N chunks of $10s$ of acquisition which corresponds to nearly 3 million points per chunk. Each recorded point cloud (chunk) is processed separately as explained in the work-flow of our pipeline presented in Figure 3, allowing a parallel processing and faster production. Yet, whereas the aforementioned filtering steps alleviate the size of the processed chunks, the resulting models remain unnecessarily heavy as flat surfaces (road, walls) may be represented by a very large number of triangles that could be drastically reduced without loosing in detail.

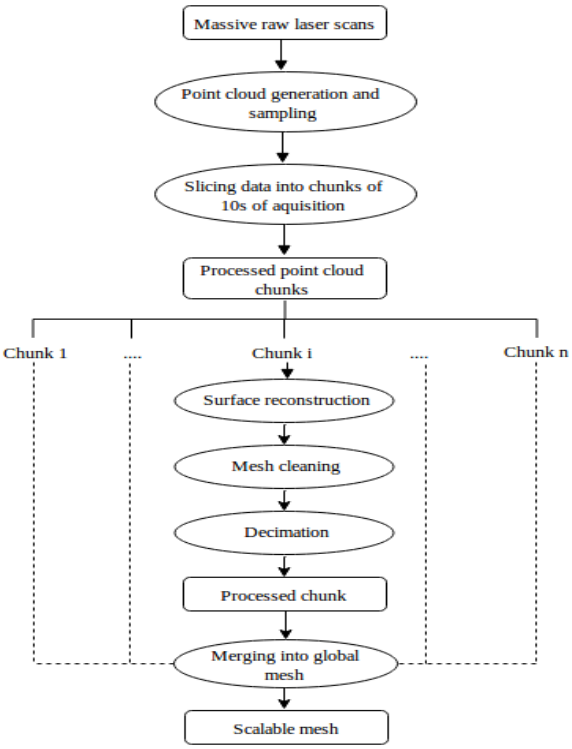


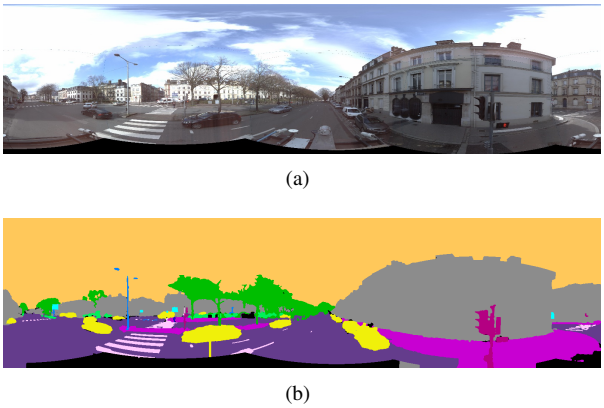
FIGURE 3 – The proposed work-flow to produce large scale models

To this end, we apply the decimation algorithm of [15, 16].

3 Semantization

Semantic segmentation has seen a rapid progress over the past decade. Recent advances achieved by different types of Convolutional Neural Networks (CNN) have improved notably the accuracy of state-of-the-art techniques [18, 19, 1]. Among the many CNN architectures available, convolutional encoder-decoder networks are particularly well adapted to the problem of pixel labeling. The encoder part of the network creates a rich feature map representing the image content and the decoder transforms the feature map into a map of class probabilities for every pixel of the input image. Such operation takes into account the pooling indices to upsample low resolution features into the original image resolution. Then, the label class with the highest probability is assigned for each pixel.

We address the semantization of the textured mesh as a problem of semantic segmentation of spherical images. The advantages of this approach is that we can exploit open source solutions like [1], and that we can benefit from transfer learning from several urban datasets [5]. Depth information can be exploited as well for semantic segmentation within this framework as shown in [7]. We have chosen to use spherical images in order to provide the maximum amount of contextual information for better semantic segmentation. The spherical images are obtained through panorama stitching of the different oriented images of the acquisition vehicle [20]. An example of semantic segmentation of a spherical image is shown in Fig. 4. The semantic labels of the images are then reprojected to the 3D mesh as explained in Section 4.2.



(a)

(b)

FIGURE 4 – Semantic segmentation of a spherical image acquired by Stereopolis-II [20].

4 Texture and label mapping

This section presents our approach for texturing and semantizing large scale 3D urban scenes. Based on the work of [31], we adapt the algorithm so it can handle our camera model and the parameters are properly adjusted to enhance the results. The aforementioned label predictions are subsequently back-projected onto the reconstructed surface yielding a semantic 3D model. In the following, we give the outline of this technique and its requirements.

To work jointly with oriented images and LiDAR scans acquired by a mobile mapping system, the first requirement is that both sensing modalities have to be aligned in a common frame. Thanks to the rigid setting of the camera and the LiDAR mounted on the mobile platform yielding a simultaneous image and LiDAR acquisition, this step is no more required. However, such setting entails that a visible part of the vehicle appears in the acquired images. To avoid using these irrelevant parts, an adequate mask is applied to the concerned images (back and front images) before tex-

turing.
Typically, texturing a 3D model with oriented images is a two-stage process. First, the optimal view per triangle is selected with respect to certain criteria yielding to a preliminary texture. Second, a color optimization is performed to minimize the discontinuities between adjacent texture patches. The two steps are discussed in Section 4.1.

4.1 View selection and color adjustment

To determine the visibility of faces in the input images, a pairwise Markov random field energy formulation is adopted to compute a labeling l that assigns a view l_i to be used as texture for each mesh face F_i :

$$E(l) = \sum_{F_i \in Faces} E_d(F_i, l_i) + \sum_{F_i, F_j \in Edges} E_s(F_i, F_j, l_i, l_j) \quad (1)$$

where

$$E_d = -\int_{\phi(F_i, l_i)} ||\nabla(I_{l_i})||_2 dp \quad (2)$$

$$E_s = [l_i \neq l_j] \quad (3)$$

The data term E_d (2) computes the gradient magnitude $||\nabla(I_{l_i})||_2$ of the image into which face F_i is projected using a Sobel operator and sum over all pixels of the gradient magnitude image within face F_i's projection $\phi(F_i, l_i)$. The absolute value of this term is large if the projection area is large which means that it prefers close, orthogonal and in-focus images with high resolution. The smoothness term E_s (3) minimizes the seams visibility (edges between faces textured with different images). In the chosen method, this regularization term is based on the Potts model which prefers compact patches without favoring distant views and it is extremely fast to compute. Finally, $E(l)$ (1) is minimized with graph-cuts and α-expansion [3].
After the view selection step, the obtained model exhibits strong color discontinuities due to the fusion of texture patches coming from different images and to the exposure and illumination variation especially in an outdoor environment. Thus, adjacent texture patches need to be photometrically adjusted. To address this problem, first, a global radiometric correction is performed along the seam's edge by computing a weighted average of a set of samples (pixels sampled along the discontinuity's right and left) depending on the distance of each sample to the seam edge extremities (vertices). Then, this global adjustment is followed by a local Poisson editing [21] applied to the border of the texture patches.
Finally, the corrections are added to the input images, the texture patches are packed into texture atlases, and texture coordinates are attached to the mesh vertices.

4.2 Multi-view optimization for 3D surface labeling

Once the semantization is performed on the spherical images, the pixel-wise class scores are projected back to the 3D mesh yielding to a texel-wise 3D semantic model. First, following the same optimization framework used for view selection explained in section 4.1, we change the data term E_d (2) to the area of the image into which face F_i is projected and we keep the smoothing term E_s (3) unchanged. The underlying assumption is that the more important the area of the view projection is, the more information is exploited by the semantization process so that it has higher confidence. Second, the photometric correction step is deactivated since there is no need to adjust the discontinuities between the borders of the semantic patches.

5 Experimental results

5.1 Mesh reconstruction and texturing

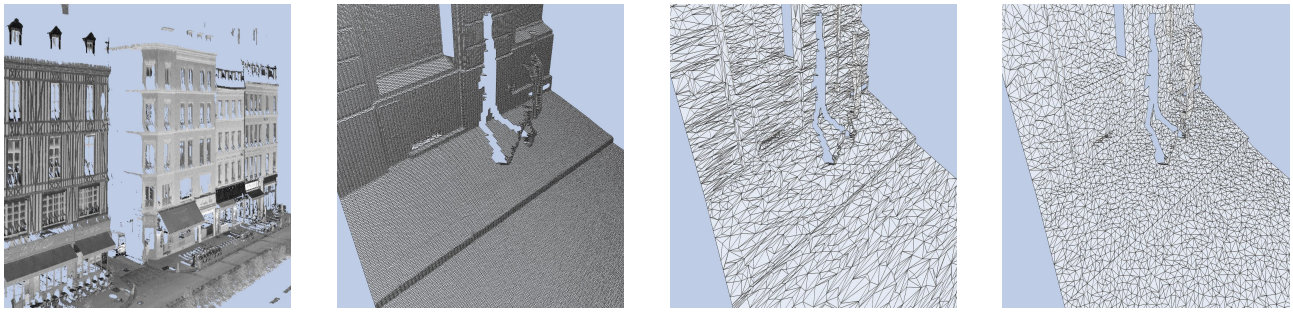
In Figure 5, we show the reconstructed mesh based on the sensor topology and the adopted decimation process. In practice, we parameterize the algorithm such that the approximation error is below 3cm, which allows in average to reduce the number of triangles to around 30% of the input triangles. Figure 6 exhibits some texturing results in different places in Rouen, France.

5.2 Semantic labeling

Our semantic segmentation model has been pre-trained with a large urban dataset [5], and has been fine-tuned later with 26 spherical images scattered along the area that aim to reconstruct. The main interest to perform the fine-tuning is to adapt the model trained with perspective images in order to work with spherical ones. An overview of the segmentation accuracy is shown in Figure 7.
In Figure 6 we show the 3D labeling results of some textured chunks in different places in Rouen, France. Despite the complexity of labeling certain classes in a 3D mesh (such as Road-marks and Pedestrian), we are able to achieve acceptable results. However, due to the fact that these classes can not be well represented in a 3D mesh unless we use hand-crafted methods to detect and properly reconstruct them, a problem arguably as hard as semantic segmentation, the semantization technique fails to assign correctly the corresponding label to each of these items because of the inconsistency between the semantic labels of the photometric modality and the 3D geometry during the back-projection from the 2D labeled images to the 3D model.

5.3 Performance evaluation

We evaluate the performance of the on-line steps of our pipeline which are the surface reconstruction and the view selection (for texturing and labeling the 3D model) on a dataset acquired by Stereopolis II [20] during this project. It consists of 17km of 6 hours of acquisition of both LiDAR and images yielding nearly to 2 billion georeferenced points and 40000 full HD images (more than 500 Gigabytes of raw data).
In Table 1, we present the required input data to texture a chunk of acquisition ($10s$); the average number of views and the number of triangles after decimation. Figure 8



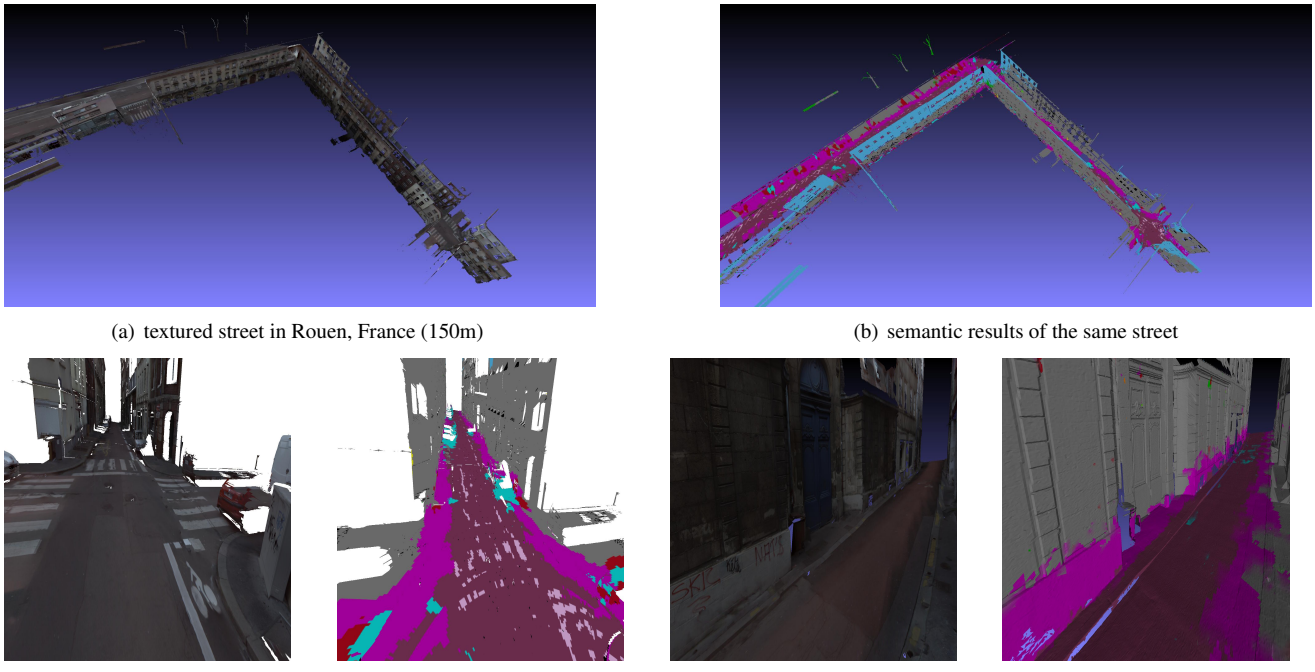
(a) the reconstructed mesh colored with reflectance

(b) zoomed region before decimation

(c) Decimated region (70%) with elongated triangles

(d) Decimated region (70%) with regular triangles

FIGURE 5 – Decimation of sensor space topology mesh



(a) textured street in Rouen, France (150m)

(b) semantic results of the same street

(c) zoomed textured region 1

(d) the corresponding semantic labeling

(e) zoomed textured region 2

(f) the the corresponding semantic labeling

FIGURE 6 – Illustration of semantic and textured parts in the acquired dataset Rouen, France (Best viewed on the screen)

Acquisition	# Views	# Faces	Image resolution
10s	120	1.8 Million	2048×2048

TABLE 1 – Statistics on the input data per chunk

shows the timing of each step in the pipeline to texture the described setting. Using a 16-core Xeon E5-2665 CPU with 12GB of memory, we are able to generate a 3D mesh of nearly 6 Million triangles in less than one minute compared to the improved version of Poisson surface reconstruction [12] where they reconstruct a surface of nearly 20000 triangle in 10 minutes. Moreover, in order to texture small models with few images (36 of size (768×584)) in a context of super-resolution, [9] takes several hours (partially on GPU) compared to the few minutes we take to texture our huge models. Finally, all the dataset can be textured or labeled in less than 30 computing hours.

6 Conclusion

This paper has demonstrated a full pipeline to produce textured and semantic 3D mesh from mobile mapping images and LiDAR data at city scale. It is mostly based on state of the art techniques that have gained a level of maturity compatible with such large scale processing. Converting the problem of semantic segmentation of a 3D scene to 2D has certainly simplified this issue. However, such method entails a loss of information, thus a limited discrimination performance. In the future we are interested in directly applying deep learning techniques to the 3D scenes which is much more promising. We believe that such a representa-

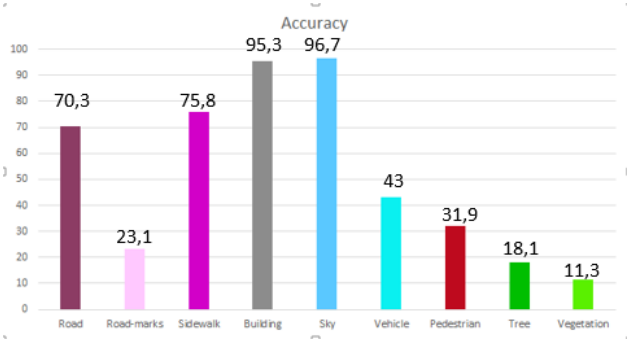


FIGURE 7 – Semantic segmentation class-wise accuracy on the test images.

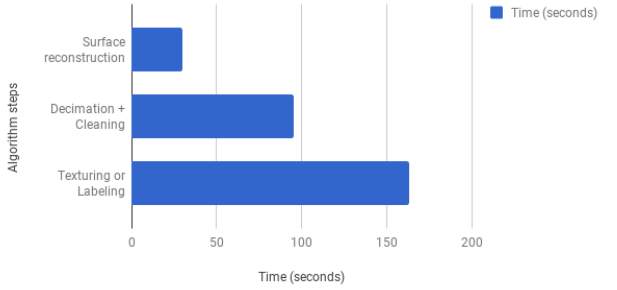


FIGURE 8 – Performance evaluation of a chunk of 10s of acquisition

tion can find multiple applications, directly through visualization of a mobile mapping acquisition, or more indirectly for robotics applications (localization, navigation…).

Acknowledgement

We would like to acknowledge the French ANR project pLaTINUM (ANR-15-CE23-0010) for its financial support.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet : A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv :1511.00561*, 2015.

[2] Alexandre Boulch, Bertrand Le Saux, and Nicolas Audebert. Unstructured point cloud semantic labeling using deep segmentation networks. In *Eurographics Workshop on 3D Object Retrieval, 3DOR 2017, Lyon, France, April 23-24, 2017*, 2017.

[3] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11) :1222–1239, 2001.

[4] Marco Callieri, Paolo Cignoni, Massimiliano Corsini, and Roberto Scopigno. Masked photo blending : Mapping dense photographic data set on high-resolution sampled 3d models. *Computers & Graphics*, 32(4) :464–473, 2008.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[6] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring spatial context for 3d semantic segmentation of point clouds. In *2017 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2017, Venice, Italy, October 22-29, 2017*, pages 716–724, 2017.

[7] Eduardo Fernandez-Moral, Renato Martins, Denis Wolf, and Patrick Rives. A new metric for evaluating semantic segmentation : leveraging global and contour accuracy. In *Workshop on Planning, Perception and Navigation for Intelligent Vehicles, PPNIV17*, 2017.

[8] Ignacio Garcia-Dorado, Ilke Demir, and Daniel G. Aliaga. Automatic urban modeling using volumetric reconstruction with surface graph cuts. *Computers & Graphics*, 37(7) :896–910, 2013.

[9] Bastian Goldlücke, Mathieu Aubry, Kalin Kolev, and Daniel Cremers. A super-resolution framework for high-accuracy multiview reconstruction. *International Journal of Computer Vision*, 106(2) :172–191, 2014.

[10] L Grammatikopoulos, I Kalisperakis, G Karras, and E Petsa. Automatic multi-view texture mapping of 3d surface projections. In *Proceedings of the 2nd ISPRS International Workshop 3D-ARCH*, pages 1–6, 2007.

[11] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap : an efficient probabilistic 3d mapping framework based on octrees. *Auton. Robots*, 34(3) :189–206, 2013.

[12] Michael M. Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3) :29 :1–29 :13, 2013.

[13] Sheraz Khan, Dirk Wollherr, and Martin Buss. Adaptive rectangular cuboids for 3d mapping. In *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*, pages 2132–2139, 2015.

[14] Peter Liepa. Filling holes in meshes. In *First Eurographics Symposium on Geometry Processing, Aachen, Germany, June 23-25, 2003*, pages 200–205, 2003.

[15] Peter Lindstrom and Greg Turk. Fast and memory efficient polygonal simplification. In *Visualization*

'98, Proceedings, October 18-23, 1998, Research Triangle Park, North Carolina, USA., pages 279–286, 1998.

[16] Peter Lindstrom and Greg Turk. Evaluation of memoryless simplification. *IEEE Trans. Vis. Comput. Graph.*, 5(2) :98–115, 1999.

[17] Vadim Litvinov and Maxime Lhuillier. Incremental solid modeling from sparse structure-from-motion data with improved visual artifacts removal. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 2745–2750, 2014.

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[19] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[20] Nicolas Paparoditis, Jean-Pierre Papelard, Bertrand Cannelle, Alexandre Devaux, Bahman Soheilian, Nicolas David, and Erwann 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*, 200(1) :69–79, 2012.

[21] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3) :313–318, 2003.

[22] Marc Pollefeys, David Nistér, Jan-Michael Frahm, Amir Akbarzadeh, Philippos Mordohai, Brian Clipp, Chris Engels, David Gallup, Seon Joo Kim, Paul Merrell, C. Salmi, Sudipta N. Sinha, B. Talton, Liang Wang, Qingxiong Yang, Henrik Stewénius, Ruigang Yang, Greg Welch, and Herman Towles. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3) :143–167, 2008.

[23] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet : Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 77–85, 2017.

[24] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++ : Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5105–5114, 2017.

[25] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet : Learning deep 3d representations at high resolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6620–6629, 2017.

[26] Hayko Riemenschneider, András Bódis-Szomorú, Julien Weissenberg, and Luc J. Van Gool. Learning where to classify in multi-view semantic segmentation. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 516–532, 2014.

[27] Andrea Romanoni, Daniele Fiorenti, and Matteo Matteucci. Mesh-based 3d textured urban mapping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 3460–3466, 2017.

[28] Andrea Romanoni and Matteo Matteucci. Incremental reconstruction of urban environments by edge-points delaunay triangulation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, pages 4473–4479, 2015.

[29] Sudipta N. Sinha, Drew Steedly, Richard Szeliski, Maneesh Agrawala, and Marc Pollefeys. Interactive 3d architectural modeling from unordered photo collections. *ACM Trans. Graph.*, 27(5) :159 :1–159 :10, 2008.

[30] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud : Semantic segmentation of 3d point clouds. *CoRR*, abs/1710.07563, 2017.

[31] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color ! Large-scale texturing of 3D reconstructions. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pages 836–850. Springer International Publishing, 2014.

[32] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision, 3DV 2013, Seattle, Washington, USA, June 29 - July 1, 2013*, pages 127–134, 2013.