

LARGE SCALE TEXTURED MESH RECONSTRUCTION FROM MOBILE MAPPING IMAGES AND LIDAR SCANS

Mohamed Boussaha¹, Bruno Vallet¹, Patrick Rives²

¹ IGN/LASTIG MATIS, Université Paris Est, 73 avenue de Paris, Saint Mandé 94160, France - (mohamed.boussaha, bruno.vallet)@ign.fr
² Inria Sophia Antipolis - Méditerranée, Lagadic team, 2004 route de Lucioles - BP93, 06902 Sophia Antipolis, France - Patrick.Rives@inria.fr

Commission II, WG II/4

KEY WORDS: Urban scene, Mobile mapping, LiDAR, Oriented images, Surface reconstruction, Texturing

ABSTRACT:

The representation of 3D geometric and photometric information of the real world is one of the most challenging and extensively studied research topics in the photogrammetry and robotics communities. In this paper, we present a fully automatic framework for 3D high quality large scale urban texture mapping using oriented images and LiDAR scans acquired 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, fast and scalable 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. Finally, the algorithm proposed in (Waechter et al., 2014) is adapted to texture the reconstructed surface with the images acquired simultaneously, ensuring a high quality texture with no seams and global color adjustment. We evaluate our full pipeline on a dataset of 17 km of acquisition in Rouen, France resulting in nearly 2 billion points and 40000 full HD images. We are able to reconstruct and texture the whole acquisition in less than 30 computing hours, the entire process being highly parallel as each chunk can be processed independently in a separate thread or computer.

1. INTRODUCTION

1.1 Context

Mobile Mapping Systems (MMS) have become more and more popular to map cities from the ground level, allowing for a very interesting compromise between level of detail and productivity. Such MMS are increasingly becoming hybrid, acquiring both images and LiDAR point clouds of the environment. However, these two modalities remain essentially exploited independently, and few works propose to process them jointly. Nevertheless, such a joint exploitation would benefit from the high complementarity of these two sources of information:

- High resolution of the images vs high precision of the LiDAR range measurement.
- Passive RGB measurement vs active intensity measurement in near infrared.
- Different acquisition geometries.

In this paper, we propose a fusion of image and LiDAR information into a single representation: a textured mesh. Textured meshes have been the central representation for virtual scenes in Computer Graphics, massively used in the video games and animation movies industry. Graphics cards are highly optimized for their visualization, and they allow a representation of scenes that holds both their geometry and radiometry. Textured meshes are now gaining more and more attention in the geospatial industry as Digital Elevation Models coupled with orthophotos, which were well adapted for high altitude airborne or space-borne acquisition, are not suited for the newer means of acquisition: closer range platforms (drones, mobile mapping) and oblique imagery.

We believe that this trend will accelerate, such that the geospatial industry will have an increasing need for efficient and high quality surface reconstruction and texturing algorithms that scale up to the massive amounts of data that these new means of acquisition produce.

This paper focuses on:

- using a simple reconstruction approach based on the sensor topology
- adapting the state-of-the-art texturing method of (Waechter et al., 2014) to mobile mapping images and LiDAR scans

We are able to produce a highly accurate surface mesh with a high level of detail and high resolution textures at city scale.

1.2 Related work

In this paper we present a visibility consistent 3D mapping framework to construct large scale urban textured mesh using both oriented images and georeferenced point cloud coming from a terrestrial mobile mapping system. In the following, we give an overview of the various methods related to the design of our pipeline.

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 (Romanoni et al., 2017). In the literature, both image-based methods (Wu, 2013, Litvinov and Lhuillier, 2014, Romanoni and Matteucci, 2015) and LiDAR-based methods (Hornung et al., 2013, Khan et al., 2015) often represent the map as a point cloud or a mesh relying only on

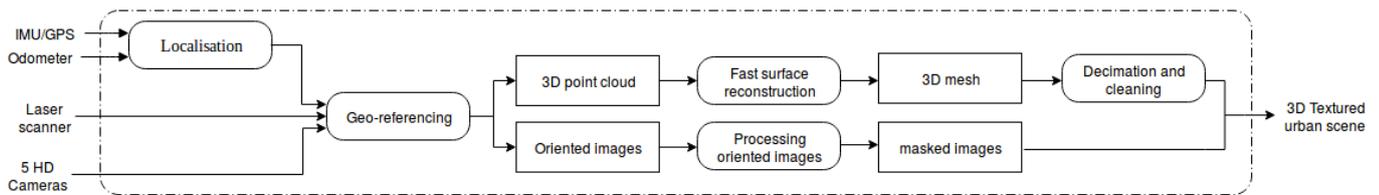


Figure 1. The texturing pipeline

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, computer graphics and photogrammetry** communities have generated compelling urban texturing results. (Sinha et al., 2008) developed an interactive system to texture architectural scenes with planar surfaces from an unordered collection of photographs using cues from structure-from-motion. (Tan et al., 2008) proposed an interactive tool for only building facades texturing using oblique images. (Garcia-Dorado et al., 2013) 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 during texturing process. In order to achieve a consistent texture across patch borders in a setting of unordered registered views, (Callieri et al., 2008, Grammatikopoulos et al., 2007) 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 burden in a reasonable time frame without increasing the geometric complexity in the final model. In contrast to the latter methods, (Waechter et al., 2014) proposed to use the multi-view stereo technique (Frahm et al., 2010, Furukawa et al., 2010) to perform a surface reconstruction and subsequently 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 the method of (Waechter et al., 2014) as a base for our work since it is the first comprehensive framework for texture mapping that enables fast and scalable processing.

In **our work**, we abstain from the surface reconstruction step for multiple reasons. As pointed out above, methods based on structure-from-motion and multi-view stereo techniques usually yield less accurate camera parameters, hence the reconstructed geometry might not be faithful to the underlying model compared to LiDAR based methods (Pollefeys et al., 2008) 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. (Caraffa et al., 2015) proposed a generic framework to generate an octree-cell based mesh and texture it with the regularized reflectance of the LiDAR. Instead, we propose a simple but fast algorithm to construct a mesh from the raw LiDAR scans and produce photo-realistic textured models. In Figure 1, we depict the whole pipeline to generate large scale high quality textured 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 to get the final model.

The rest of the paper is organized as follows: In Section 2 we present the data acquisition system. A fast and scalable mesh reconstruction algorithm is discussed in Section 3. Section 4 explains the texturing approach. We show our experimental results in Section 5. Finally, in Section 6, we conclude the paper proposing out some future direction of research.

2. DATA ACQUISITION

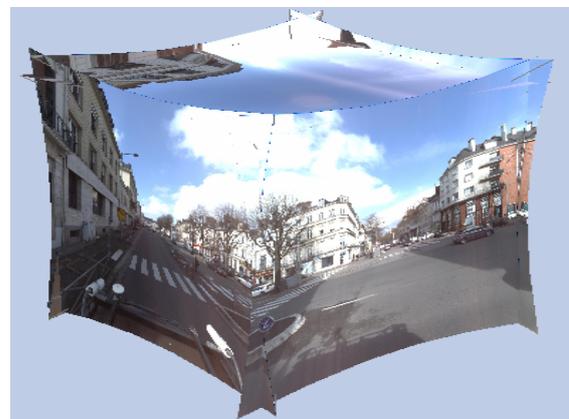


Figure 2. The set of images acquired by the 5 full HD cameras (left, right, up, in front, from behind)

Mobile mapping is a data acquisition technique that relies on sensors mounted on a terrestrial platform equipped with a georeferencing system allowing a very precise geolocalization of the acquired data. In this study, the used data was acquired by the Mobile Mapping System (MMS) Stereopolis II (Paparoditis et al., 2012) developed at the french National Mapping Agency (IGN). It is equipped with 5 full HD cameras (2048 × 2048 px) acquiring 12-bit RGB images that correspond to the five upper faces of a cube as shown in Figure 2 and a LiDAR sensor mounted transversally so it is able to scan the plane orthogonal to the vehicle trajectory.

The used LiDAR scanner is a RIEGL VQ-250 that rotates at 100 Hz and emits 3000 pulses per rotation with 0 to 8 echoes recorded

for each pulse, producing an average of 250 thousand points per second in typical urban scenes. The sensor records information for each pulse (direction (θ, ϕ) , time of emission) and echo (amplitude, range, deviation).

The MMS is also mounted with a georeferencing system combining a GPS, an inertial unit and an odometer. This system outputs the reference frame of the system in geographical coordinates at 100Hz. Combining this information with the information recorded by the LiDAR scanner and its calibration, a point cloud in (x, y, z) coordinates can be constructed. In the same way, using the intrinsic and extrinsic calibrations of each camera, each acquired image can be precisely oriented. It is important for our application to note that this process ensures that images and LiDAR points acquired simultaneously are precisely aligned (depending on the quality of the calibrations).

3. 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.

3.1 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 get an isotropic distribution of the line scans. 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.

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 indexes: 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 3. 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 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.

3.2 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

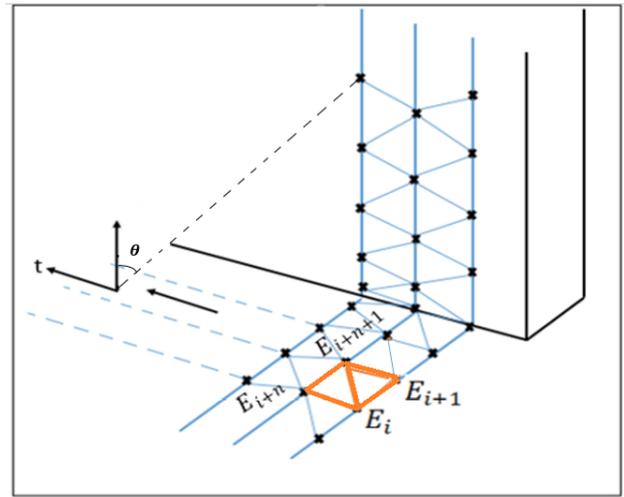


Figure 3. Triangulation based on the sensor space topology

on three main issues that frequently occur with mobile terrestrial systems and affect significantly the texturing results if not adequately dealt with.

3.2.1 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.

3.2.2 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 (set experimentally) are automatically deleted from the final model.

3.2.3 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 (Liepa et al., 2003). The algorithm takes a user-defined parameter which consists of the maximum hole size in terms of number of edges and closes 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.

3.3 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 (up to 1 TB by day of acquisition (Paparoditis et al., 2012)) 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 4, 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.

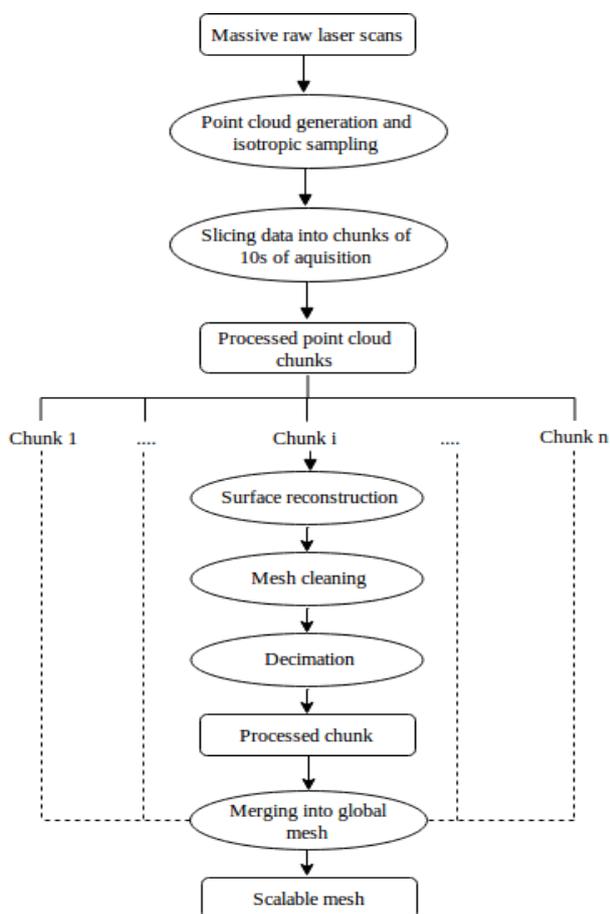


Figure 4. The proposed work-flow to produce large scale models

To this end, we apply the decimation algorithm of (Lindstrom and Turk, 1998, Lindstrom and Turk, 1999). The algorithm proceeds in two stages. First, an initial collapse cost, given by the position chosen for the vertex that replaces it, is assigned to every edge in the reconstructed mesh. Then, at each iteration the edge with the lowest cost is selected for collapsing and replacing it with a vertex. Finally, the collapse cost of all the edges now incident on the replacement vertex is recalculated. For more technical details, we refer the reader to (Lindstrom and Turk, 1998, Lindstrom and Turk, 1999).

4. TEXTURING APPROACH

This section presents the used approach for texturing large scale 3D realistic urban scenes. Based on the work of (Waechter et al., 2014), we adapt the algorithm so it can handle our camera model (with five perspective images) and the smoothing parameters are properly adjusted to enhance the results. In the following, we give the outline of this texturing technique and its requirements.

4.1 Image processing and registration



(a) before masking (b) after masking

Figure 5. Illustration of the acquired frontal images processing

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 automatically applied to the concerned images (back and front images) before texturing as shown in Figure 5.

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 a preliminary texture. Second, a local and global color optimization is performed to minimize the discontinuities between adjacent texture patches. The two steps are discussed in Sections 4.2 and 4.3.

4.2 View selection

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

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

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

The data term E_d (2) computes the gradient magnitude $\|\nabla(I_i)\|_2$ 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 $\phi(F_i, l_i)$. 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 ([1] the Iverson bracket) which prefers compact patches by penalizing those that might give severe seams in the final model and it is extremely fast to compute. Finally, $E(l)$ (1) is minimized with graph-cuts and α -expansion (Boykov et al., 2001).

4.3 Color adjustment

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 (Pérez et al., 2003) applied to the border of the texture patches. All the process is discussed in details in (Waechter et al., 2014) work.

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.

5. EXPERIMENTAL RESULTS

5.1 Mesh reconstruction

In Figure 6, 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.

5.2 Texturing the reconstructed models

In this section, we show some texturing result (Figure 7) and the influence of the color adjustment step on the final textured models (Figure 8). Before the radiometric correction, one can see several color discontinuities especially on the border of the door and on some parts of the road (best viewed on screen). More results are presented in the appendix to illustrate the high quality textured models in different places in Rouen, France.

5.3 Performance evaluation

We evaluate the performance of each step of our pipeline on a dataset acquired by Stereopolis II (Paparoditis et al., 2012) 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 Giga-bytes of raw data).

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

Table 1. Statistics on the input data per chunk

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 9 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 (Kazhdan and Hoppe, 2013) 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, (Goldlücke et al., 2014) takes several hours (partially on GPU) compared to the few minutes we take to texture our huge models. Finally, all the dataset is textured in less than 30 computing hours.

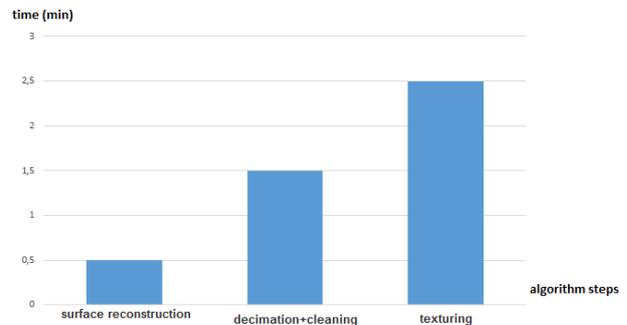


Figure 9. Performance evaluation of a chunk of 10s of acquisition

6. CONCLUSION AND PERSPECTIVES

6.1 CONCLUSION

This paper has demonstrated a full pipeline to produce textured 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. The sensor mesh reconstruction is quite novel but very simple. We believe that such a textured mesh can find multiple applications, directly through visualization of a mobile mapping acquisition, or more indirectly for processing jointly image and LiDAR data: urban scene analysis, structured reconstruction, semantization, ...

6.2 PERSPECTIVES

This work leaves however important topics unsolved, and most importantly the handling of overlaps between acquired data, at intersections or when the vehicle passes multiple times in the same scene. We have left this issue out of the scope of the current paper as it poses numerous challenges:

- Precise registration over the overlaps, sometimes referred to as the loop-closure problem.
- Change detection between the overlaps.
- Data fusion over the overlaps, which is strongly connected to change detection and how changes are handled in the final model.

Moreover, this paper proposed a reconstruction from LiDAR only, but we believe that the images hold a pertinent geometric information that could be used to complement the LiDAR reconstruction, in areas occluded to the LiDAR but not to the cameras (which often happens as their geometries are different).

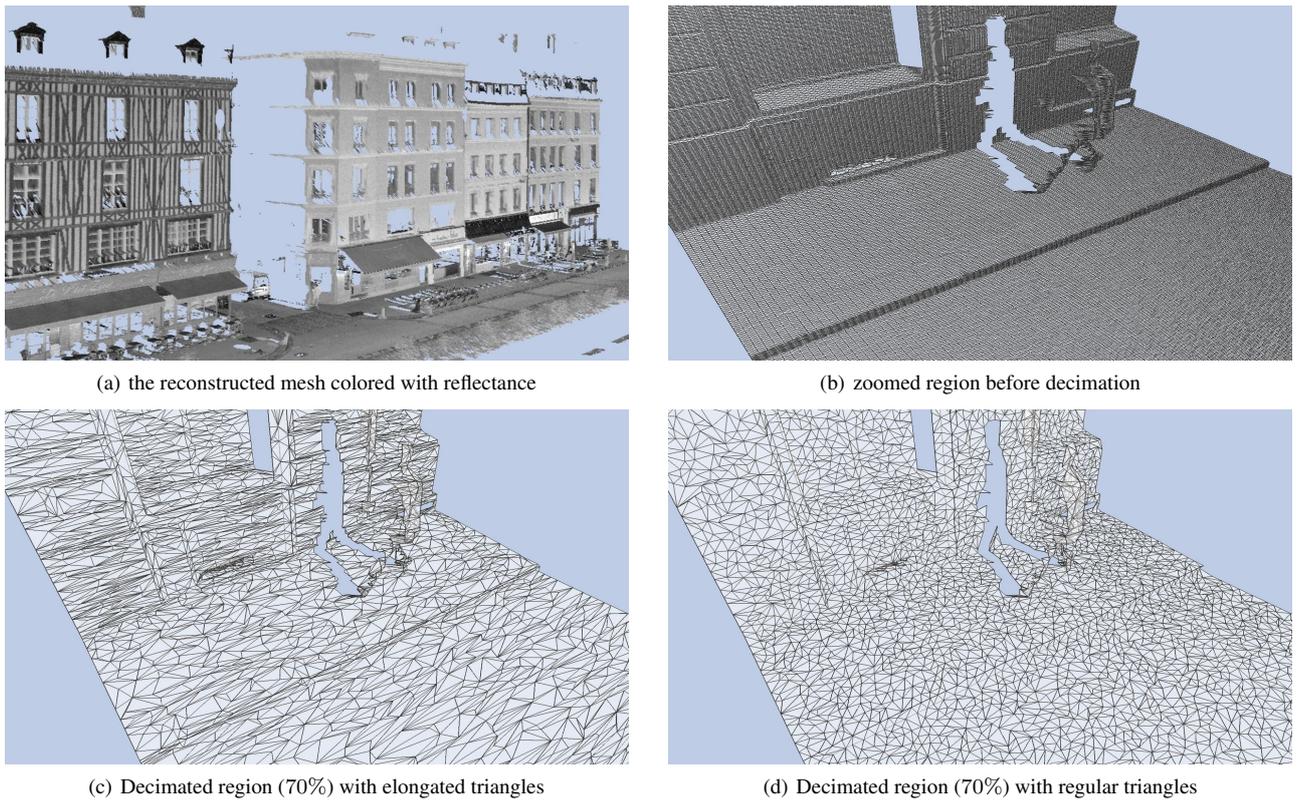


Figure 6. Decimation of sensor space topology mesh



Figure 7. Texturing result on a region in Rouen, France

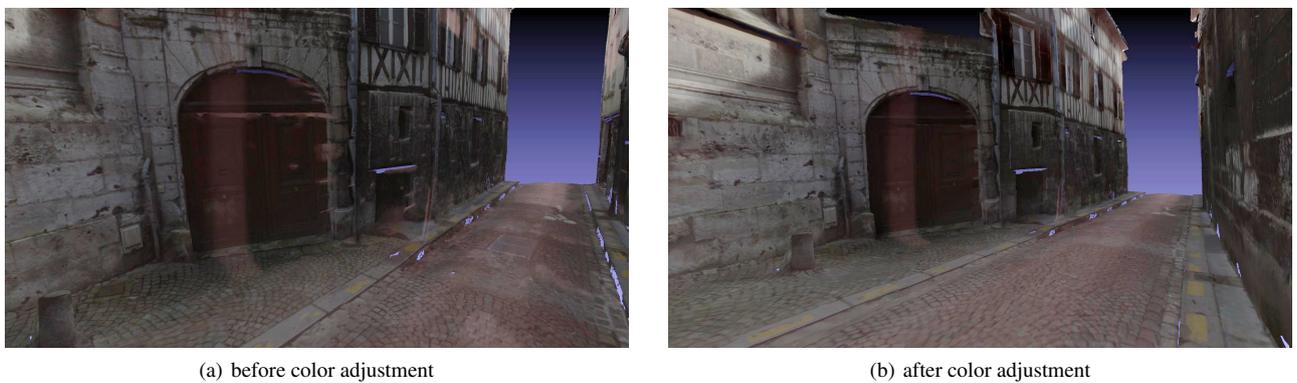


Figure 8. Textured scene before (a) and after color adjustment (b)

Finally, an important issue that is partially tackled in the texturation: the presence of mobile objects. Because the LiDAR and images are most of the time not acquired strictly simultaneously, mobile objects might have an incoherent position between image and LiDAR, which is a problem that should be tackled explicitly.

ACKNOWLEDGEMENTS

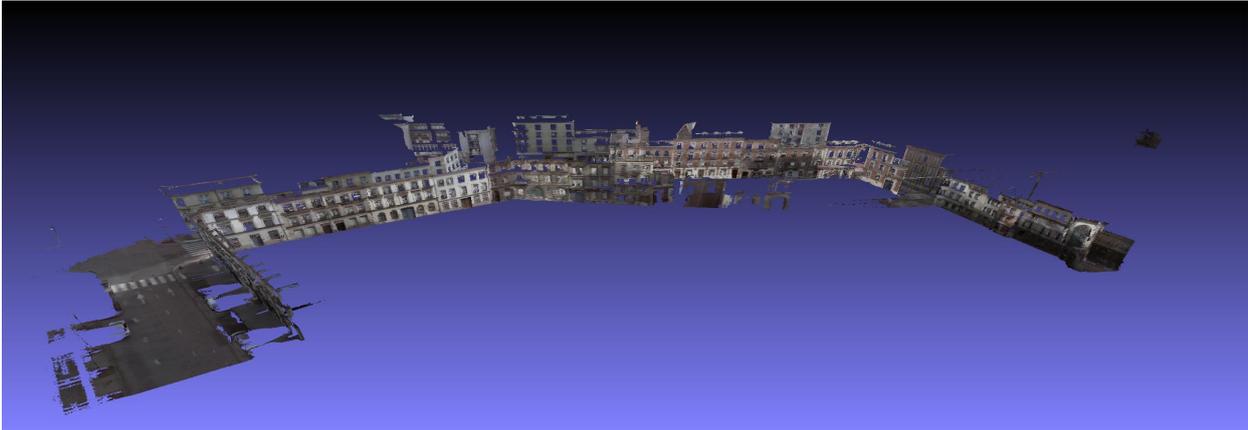
We would like to acknowledge the French ANR project pLaT-INUM (ANR-15-CE23-0010) for its financial support.

REFERENCES

- Boykov, Y., Veksler, O. and Zabih, R., 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* 23(11), pp. 1222–1239.
- Callieri, M., Cignoni, P., Corsini, M. and Scopigno, R., 2008. Masked photo blending: Mapping dense photographic data set on high-resolution sampled 3d models. *Computers & Graphics* 32(4), pp. 464–473.
- Caraffa, L., Brédif, M. and Vallet, B., 2015. 3d octree based watertight mesh generation from ubiquitous data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40(3), pp. 613.
- Frahm, J., Georgel, P. F., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y., Dunn, E., Clipp, B. and Lazebnik, S., 2010. Building rome on a cloudless day. In: *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, pp. 368–381.
- Furukawa, Y., Curless, B., Seitz, S. M. and Szeliski, R., 2010. Towards internet-scale multi-view stereo. In: *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pp. 1434–1441.
- Garcia-Dorado, I., Demir, I. and Aliaga, D. G., 2013. Automatic urban modeling using volumetric reconstruction with surface graph cuts. *Computers & Graphics* 37(7), pp. 896–910.
- Goldlücke, B., Aubry, M., Kolev, K. and Cremers, D., 2014. A super-resolution framework for high-accuracy multiview reconstruction. *International Journal of Computer Vision* 106(2), pp. 172–191.
- Grammatikopoulos, L., Kalisperakis, I., Karras, G. and Petsa, E., 2007. Automatic multi-view texture mapping of 3d surface projections. In: *Proceedings of the 2nd ISPRS International Workshop 3D-ARCH*, pp. 1–6.
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C. and Burgard, W., 2013. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Auton. Robots* 34(3), pp. 189–206.
- Kazhdan, M. M. and Hoppe, H., 2013. Screened poisson surface reconstruction. *ACM Trans. Graph.* 32(3), pp. 29:1–29:13.
- Khan, S., Wollherr, D. and Buss, M., 2015. Adaptive rectangular cuboids for 3d mapping. In: *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*, pp. 2132–2139.
- Lindstrom, P. and Turk, G., 1998. Fast and memory efficient polygonal simplification. In: *Visualization '98, Proceedings, October 18-23, 1998, Research Triangle Park, North Carolina, USA.*, pp. 279–286.
- Lindstrom, P. and Turk, G., 1999. Evaluation of memoryless simplification. *IEEE Trans. Vis. Comput. Graph.* 5(2), pp. 98–115.
- Litvinov, V. and Lhuillier, M., 2014. 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*, pp. 2745–2750.
- Paparoditis, N., Papelard, J.-P., Cannelle, B., Devaux, A., Soheilian, B., David, N. and Houzay, E., 2012. Stereopolis ii: A multi-purpose and multi-sensor 3d mobile mapping system for street visualisation and 3d metrology. *Revue française de photogrammétrie et de télédétection* 200(1), pp. 69–79.
- Pérez, P., Gangnet, M. and Blake, A., 2003. Poisson image editing. *ACM Trans. Graph.* 22(3), pp. 313–318.
- Pollefeys, M., Nistér, D., Frahm, J., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S. J., Merrell, P., Salmi, C., Sinha, S. N., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G. and Towles, H., 2008. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision* 78(2-3), pp. 143–167.
- Romanoni, A. and Matteucci, M., 2015. 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*, pp. 4473–4479.
- Romanoni, A., Fiorenti, D. and Matteucci, M., 2017. 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*, pp. 3460–3466.
- Sinha, S. N., Steedly, D., Szeliski, R., Agrawala, M. and Pollefeys, M., 2008. Interactive 3d architectural modeling from unordered photo collections. *ACM Trans. Graph.* 27(5), pp. 159:1–159:10.
- Tan, Y., Kwoh, L. and Ong, S., 2008. Large scale texture mapping of building facades. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.
- Waechter, M., Moehrle, N. and Goesele, M., 2014. Let there be color! Large-scale texturing of 3D reconstructions. In: D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars (eds), *ECCV 2014, Lecture Notes in Computer Science*, Vol. 8693, Springer International Publishing, pp. 836–850.
- Wu, C., 2013. Towards linear-time incremental structure from motion. In: *2013 International Conference on 3D Vision, 3DV 2013, Seattle, Washington, USA, June 29 - July 1, 2013*, pp. 127–134.

APPENDIX

In this appendix, we show more texturing results obtained from the acquired data in Rouen, France. Due to memory constraints, we are not able to explicitly show the entire textured model (17Km). However, we can show a maximum size of 70s (350m) of textured acquisition (Figure 10).



(a) 350 m textured scene (70s of acquisition)



(d) zoomed textured street walls

Figure 10. Samples of textured scenes from Rouen, France
This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
<https://doi.org/10.5194/isprs-annals-IV-2-49-2018> | © Authors 2018. CC BY 4.0 License.