Generating compact meshes under planar constraints: an automatic approach for modeling buildings from aerial LiDAR

Yannick Verdie, Florent Lafarge, Josiane Zerubia

GENERATING COMPACT MESHES UNDER PLANAR CONSTRAINTS: AN AUTOMATIC APPROACH FOR MODELING BUILDINGS FROM AERIAL LIDAR

Yannick Verdié, Florent Lafarge, Josiane Zerubia

Ariana Research Group - INRIA Sophia Antipolis Méditerranée, France
Firstname.Name@inria.fr

ABSTRACT

We present an automatic approach for modeling buildings from aerial LiDAR data. The method produces accurate, watertight and compact meshes under planar constraints which are especially designed for urban scenes. The LiDAR point cloud is classified through a non-convex energy minimization problem in order to separate the points labeled as *building*. Roof structures are then extracted from this point subset, and used to control the meshing procedure. Experiments highlight the potential of our method in term of minimal rendering, accuracy and compactness.

***Index Terms*—** 3D-Modeling, Shape analysis, Mesh processing, LiDAR, Graph-cuts.

1. INTRODUCTION

1.1. Related works on building reconstruction

Three dimensional building reconstruction is a topic of major interest with various applications such as urban planning, video games, virtual tour, and so on. Indeed, many works are devoted to automatically process aerial and terrestrial data *e.g* [1, 2, 3, 4]. Aerial LiDAR data processing is a mean to obtain an accurate and automatic modeling of cities. Two main trends are discerned in the literature. Some works generate meshes from the point clouds using geometry processing, *e.g.* [5]. However, the different urban components are difficult to identify in such mesh-based representations, and the resulting mesh is not necessary compact. Other works intend to outperform the limitation of a mesh based approach by considering geometrical primitives [6, 7]. While this family of approaches is based on strong urban knowledge, it is mainly restricted by the set of primitives of the considerate model and the final representation may not be as accurate as a mesh-based reconstruction. A more general approach proposes to parse a building into a high level hierarchical representation [8].

1.2. Contributions

Our work presents innovating solutions in the field by taking advantage of the mesh-based representations as well as the primitive-based approach, and providing both semantic information in the 3D-representation and a realistic reconstruction. We focus on reconstructing the buildings and do not consider the other urban elements which are excluded by the point cloud classification process. We shall demonstrate that our results use few vertices while remaining accurate. We adopt a three-step strategy illustrated on Fig. 1. First of all, the point cloud is classified by defining a non-supervised minimization problem as presented in Section 2. The second step, detailed in Section 3, consists in extracting meaningful building structures from the points labeled as *building*. Finally, the generation of a compact mesh preserving the building structures is presented in Section 4. Experimental results are shown and commented in Section 5.
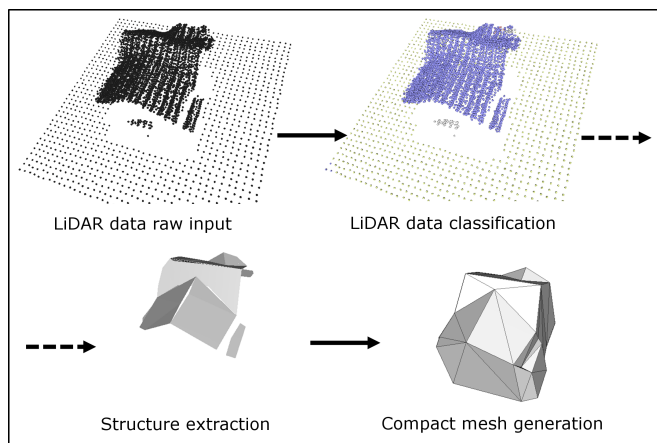

Fig. 1. Pipeline of the proposed approach.

2. POINT CLOUD CLASSIFICATION

The first step consists in classifying the input point cloud in four classes: *building*, *vegetation*, *ground*, and *Not-Clustered* (*NC*). A Digital Terrain Model (*DTM*) is computed based on a historical analysis on the height point component. Four geometric attributes are computed for each point. $Elevation$ a_e is the height difference between the *DTM* and the point height component. $Local$ $Planarity$ a_p is the quadratic distance between the point and the least squared 3D plane computed with its spherical neighborhood. $Scatter$ a_s is the local height dis-

person of the point. *Local Linearity* a_l is the quadratic distance between the point and the least squared 3D line computed with its spherical neighborhood. Those attributes are weighted by four coefficients, α_e, α_p, α_s and α_l and then normalized in the interval [0,1] as follow:

$$\begin{aligned} \beta_e &= \min\{1, \frac{|a_e|}{\alpha_e}\} \,, \beta_p = \min\{1, \frac{|a_p|}{\alpha_p}\} \\ \beta_s &= a_s^{\alpha_s} \,, \beta_l = \max\{\mathbf{1}\{a_p < \alpha_p\}, \beta_p\} \end{aligned} \quad (1)$$

A non-supervised energy minimization problem is defined to classify the point cloud. Graph-cuts with α-expansion [9] is used to reach a solution close to the global optimum of the energy defined as:

$$E(x) = \sum_i D_i(x_i) + \gamma \sum_{\substack{i=1..n \\ j \in N_i}} \delta(x_i, x_j) \quad (2)$$

where $\delta(x_i, x_j)$ is the pairwise interaction between the label x_i of the point i and the label x_j of the point j, defined as the standard Potts model with γ parameter. N_i is the neighboring points of the point i and $D_i(x_i)$ is the partial data term defined as a combination of the four weighted geometrical attributes:

$$D_i(x_i) = \begin{cases} \beta_e.\beta_p.\beta_s & if \quad x_i = ground \\ (1-\beta_e).(1-\beta_p).(1-\beta_s) & if \quad x_i = vegetation \\ (1-\beta_p).\beta_s.\beta_l & if \quad x_i = NC \\ (1-\beta_e).\beta_p.\beta_s & if \quad x_i = building \end{cases} \quad (3)$$

The initial configuration is chosen as $\arg_x \min \sum_i D_i(x_i)$. One can see that our energy function fits the requirements for using Graph-cuts algorithm. The weight parameters α_e, α_p, α_s and α_l were set to 6, 0.5, 0.05 and 0.25 respectively. Those parameters could be improved by a learning method, but we notice a stable behavior of our system for a wide range of data and do not think it is necessary to make the system heavier. The resulting classification is illustrated on Fig. 2.
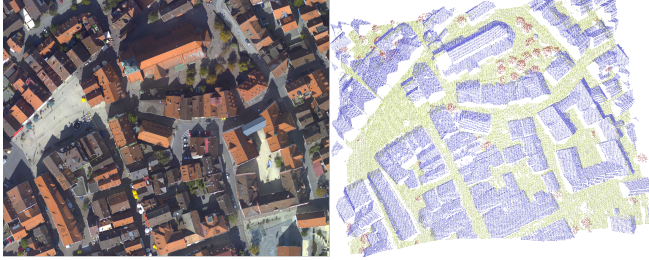

Fig. 2. Point cloud classification- An aerial image (*left*) and the classified point set (*right*). Note that the four classes *building* (in blue), *vegetation* (in red), *ground* (in green), and *NC* (in white) are correctly separated.

3. BUILDING STRUCTURE EXTRACTION

The second step consists in extracting some structural information related to the buildings .

Building footprints- They are extracted by projecting the points labeled as *building* and *NC* on a XY-grid, and then by locally propagating the information on the empty cells of the grid. As the building facades are located in between the points labeled as *ground* and the points labeled as *building* or *NC*, the building boundary is dilated by one cell-size.

Roof sections- The *planar components of the roofs* are detected by region growing. One could decide to also extract non-planar elements using an iterative non-linear minimization such as Levenberg-Marquardt optimization, but this solution was not retained because of its expensive CPU cost. Indeed, a non-planar roof can be approximated by a side-by-side sequence of small planar clusters with a satisfactory result in most cases. The propagation of the region growing is based on the regularity of the point normals in a local neighborhood.

The *roof contours* are located through a sequence of connected 3D- segments. The points representing the building edges are detected using the 2D Alpha-shape algorithm [10]. Indeed, this method is a generalization of the convex hull method and obtains better results than the original convex hull because the shape of each roof is not necessary convex. However, it requires an additional parameter α such as each edge of the resulting contour has a circle of radius α empty. The triangulation of the 2D point cloud (only XY value of each point is considered) is computed to determine a set of possible α-shapes. As the point cloud density in our experiments is about 2 points per square meter, the first α-shape with $\alpha > 3$ is considered. The contour is then simplified using an adaptive Douglas-Peucker algorithm [10]. The parameter ϵ, which is required to stop the procedure, constitutes a crucial point because it controls the simplification level of the contours. It is computed by taking into account the size of the contour:

$$\epsilon = \epsilon_{Min} + (\epsilon_{Max} - \epsilon_{Min}).(1 - e^{-\frac{nb}{RC}}) \quad (4)$$

where nb is the number of points of the contour, ϵ_{Min} and ϵ_{Max} are respectively the lower and upper bounds fixed to 0.03 and 0.3, and RC is the mean number of points fixed to 50. One can understand that a small contour requires more details (low ϵ) whereas a large contour does not (high ϵ).

Finally, some *structure adjustments* are made on the roof sections in order to connect them when they are close enough. A neighborhood relationship between roof sections is defined: two roof components are neighbors if their Euclidean distance is inferior to one meter. For each pair of neighboring roof sections, the roof contour points of interest are then projected on the line intersecting the two planes. While this step may be sufficient for simple building (e.g. building #1 on Fig. 5) , it does not guaranty a watertight reconstruction.

4. COMPACT MESH GENERATION

The last stage allows us to generate a watertight and compact mesh from the structural information extracted previously.

Mesh initialization- An initial mesh is generated from the regular XY-grid used for computing the building footprints. The cells of the grid are labeled as *interior*, *dilated*, or *boundary* depending on whether the cell belongs to the inside, the dilated, or the boundary sectors of the building footprints (see Fig. 3). For each *interior* or *dilated* cell, we test whether the cell center is located inside the planimetrical projection of the roof sections extracted in Section 3. Two cases have to be distinguished:

- The cell center belongs to one or more roof sections: For each concerned roof section, the cell center is vertically projected on the 3D-plane to create a potential vertex. The vertex having the highest Z value is kept to generate the mesh.
- The cell center does not belong to a roof section: The vertex is computed as the vertical projection of the cell center to the closest roof section. However, only *interior* cells are considered here in order to contain the propagation. This case allows to fill the holes inside the mesh without changing its topology.
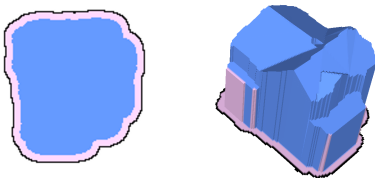

Fig. 3. Mesh initialization - Building footprint (*left*) and initial mesh (*right*). The three classes *interior*, *dilated* and *boundary* are respectively represented in blue, pink and black.

Mesh simplification with topology preservation- A quadric edges collapse decimation based algorithm [11] using topological preservation constraints is iteratively used to efficiently reduce the size of the mesh. This process preserves the mesh topology by using a quadratic error measure for the cost function and by minimizing the new location error for the placement function. Thus, planar based structures of the building are preserved during the mesh simplification. The decimation process is stopped when the number of not-*interior* vertices falls down under 1% of its initial number. As a post-process step, the remaining facets from the *boundary* are deleted. The result is illustrated on Fig. 1.

5. EXPERIMENTS AND DISCUSSION

Our method has been tested on different point cloud densities. Fig 5 presents results obtained from various types of buildings and dense urban areas.

Visual considerations- The method proposes convincing 3D-models in which the building structure is correctly retrieved. As illustrated on the Fig 5 result #1, the building components are optimally represented by the mesh in terms

of vertices and facets: each rectangular facade of the building is described by two complementary triangular facets, as well as each roof section is modeled by one or two facets. The non-planar roof sections are approximated by piecewise planar shapes: the visual rendering is not always satisfactory, as shown on the result #6 with the spherical roof.

Compactness and accuracy assessments- Our reconstruction is compared with the result obtained by [5] in Fig. 4. Our mesh has a better compactness (43 vertices / 69 facets VS 133 vertices / 228 facets) while having similar accuracy. Indeed, the average Hausdorff distance of our mesh with respect to the reconstruction by [5] is 0.16 meter. However, one can notice that a small chimney has been omitted in our reconstruction. The Hausdorff distance is maximal at this location, *i.e.* 1.12 meter. Our method focuses on the roof section reconstruction and is not designed for detecting the small planar components such as dormer-windows and chimneys. Despite those small omitted structures, the average error is very satisfactory, particularly considering the gain in terms of compactness. Moreover, to make fair comparaison with the result obtained by [5], we have chosen to use their method to generate a new mesh with similar average Hausdorff error of 0.16 meter. The mesh obtained still has a lower compactness (76 vertices / 124 facets) compared to our result.
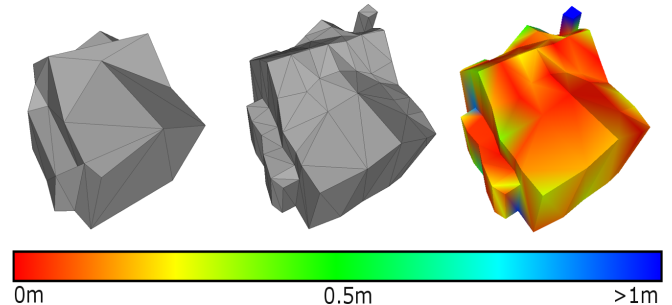

Fig. 4. Compactness and accuracy evaluation- our mesh (*left*), mesh by Zhou et al. [5] (*middle*), and the Hausdorff distance of our mesh with respect to the reconstruction by [5] (*right*). Color scale: red (no error) to blue (error superior to 1 meter).

Performances- The computing times are reasonable thank to the Computational Geometry Algorithms Library (CGAL) [12] used to performed our 3D-geometry operations. The result #8 on Fig. 5 representing a dense urban area (raw input: 139983 points) has been obtained in 9 minutes. The resulting mesh has 3450 vertices and 5788 facets.

Limitations- Our method has some limitations. The heuristic used for the Quadric edge decimation stop criteria provides good results in most cases, but it may also remove too many edges. Moreover, as we can see on Fig. 5, a sequence of small planar roofs is not alway well suited for some non-planar roofs (*e.g* the church tower on result #2 and the spherical building on result #6).
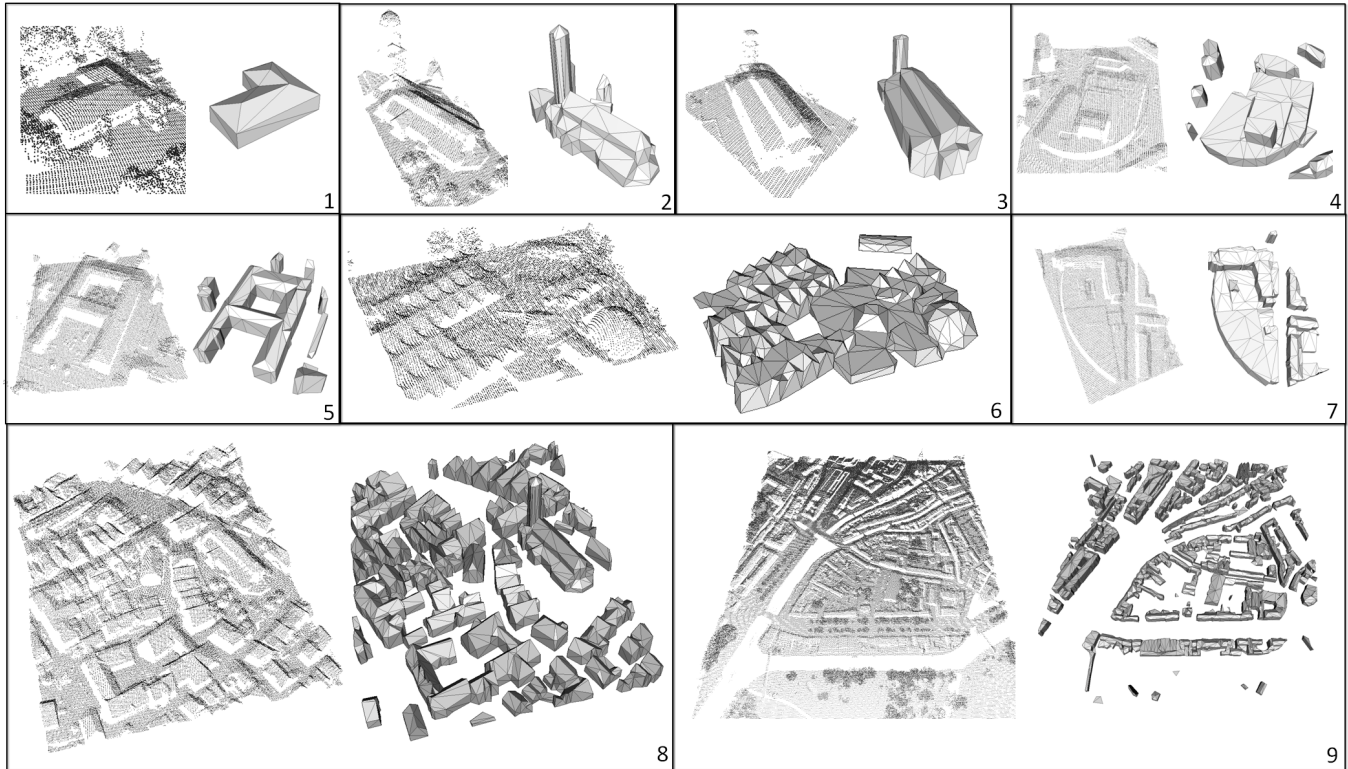
Fig. 5. Reconstruction of various buildings and urban areas- LiDAR point cloud (*left*), and our mesh-based model (*right*). Comments: see text.

6. CONCLUSION

We propose an original framework for reconstructing buildings from aerial liDAR. Our method offers an accurate representation, and performs quickly on wide ranges of data. The resulting mesh has its vertices labeled as *interior*, *boundary* or *dilated*, and is associated with the roof structures used to generated them. This semantic provides additional information that could be useful for texturing, urban planing, and so on. Finally, our mesh-based model is extremely compact and has good accuracy. As a future work, it would be interesting to use a mesh decimation scheme adapting more accurately the stop criteria to the type of building. Moreover, buildings with non-planar structures may not be well represented with our current method and working on non-planar structure extraction would improve the results. Another interesting idea would be to substitute the regular XY-grid of our method by a quadtree in order to reduce the computation times.

7. ACKNOWLEDGMENT

8. REFERENCES

[1] L. Zebedin, J. Bauer, K. F. Karner, and H. Bischof, "Fusion of feature- and area-based information for urban buildings modeling from aerial imagery," in *ECCV*, 2008.

[2] C. Früh and A. Zakhor, "An automated method for large-scale, ground-based city model acquisition," *IJCV*, vol. 60, pp. 5–24, October 2004.

[3] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, "Building reconstruction from a single dem," in *CVPR*, 2008.

[4] Z.G. Zhu and T. Kanade, "Modeling and representations of large-scale 3d scenes: Introduction to special issue," *IJCV*, vol. 78, no. 2-3, 2008.

[5] Q. Zhou and U. Neumann, "2.5d dual contouring: A robust approach to creating building models from aerial lidar point clouds," in *ECCV*, 2010.

[6] C. Poullis and S. You, "Automatic reconstruction of cities from remote sensor data," in *CVPR*, 2009.

[7] B.C. Matei, H.S. Sawhney, S. Samarasekera, J. Kim, and R. Kumar, "Building segmentation for densely built urban regions using aerial lidar data," in *CVPR*, 2008.

[8] A. Toshev, P. Mordohai, and B. Taskar, "Detecting and parsing architecture at city scale from range data," in *CVPR*, 2010.

[9] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. PAMI*, vol. 23, no. 11, pp. 1222–1239, 2001.

[10] J.R. Sack and J. Urrutia, *Handbook of computational geometry*, Elsevier, 2000.

[11] M. Garland and P. Heckber, "Surface simplification using quadric error metrics," in *SIGGRAPH*, 1997.

[12] "CGAL," http://www.cgal.org.