CITY RECONSTRUCTION FROM AIRBORNE LIDAR : A COMPUTATIONAL GEOMETRY APPROACH

Jean-Philippe Bauchet [1, 2]*, Florent Lafarge [2]

[1] LuxCarta Technology, Mouans-Sartoux, France - jean-philippe.bauchet@inria.fr
[2] Inria, Sophia-Antipolis, France - florent.lafarge@inria.fr

KEY WORDS: Urban reconstruction, airborne Lidar, point clouds, classification, kinetic data structures

ABSTRACT:

We introduce a pipeline that reconstructs buildings of urban environments as concise polygonal meshes from airborne LiDAR scans. It consists of three main steps : classification, building contouring, and building reconstruction, the two last steps being achieved using computational geometry tools. Our algorithm demonstrates its robustness, flexibility and scalability by producing accurate and compact 3D models over large and varied urban areas in a few minutes only.

1. INTRODUCTION

Urban reconstruction techniques have attracted an increasing attention from the scientific community over the last decade. Applications of such techniques include, for instance, urban planning, natural disaster management, or radio-wave propagation (Biljecki et al. 2015).

Various data sources are considered by such algorithms, such as ground imagery, satellite imagery, or pre-existing GIS data. However, despite high acquisition costs, LiDAR point clouds are more accurate and remain a data source commonly used by land surveying offices or civil engineers (Musialski et al. 2013).

Yet, models generated by current city modeling techniques may be enriched with semantic information, or represented using progressive levels of detail. The CityGML standard (Gröger et al. 2012) considers four coarse-to-fine levels applicable to airborne datasets, from LOD0 to LOD3.

In this paper, we present a pipeline that receives as input an unstructured point set describing a urban environment, and generates as output a LOD2 representation of the scene in compliance with the CityGML standard. More precisely, we aim at providing a faithful reconstruction of buildings with tilted roofs. The representation of superstructures like dormers or chimneys is beyond the scope of this paper. Our main idea is to make use of powerful computational geometry tools to extract the geometric signatures of the observed buildings. We emphasize the accuracy and the scalability of our method, since it is able to process large datasets with millions of points, dense or sparse, in a few minutes.

2. RELATED WORKS

There exists a vast literature on automatic urban reconstruction techniques, (Haala and Kada 2010, Rottensteiner et al. 2014, Musialski et al. 2013), demonstrating the deep interest of scientists and industrials for this research topic. Various data sources may be considered, leading us to draw a first distinction between all existing approaches, based on this criterion. Indeed, some algorithms may specifically address the problem

of large-scale urban reconstruction from aerial imagery (Zebedin et al. 2008, Zeng et al. 2018), satellite imagery (Duan and Lafarge 2016) or multi-view stereo dense meshes (Zhu et al. 2018, Verdie et al. 2015, Salinas et al. 2015). Some others combine different sources of data to generate urban models with the finest level of detail (Kelly et al. 2017).

In this work, we focus on the problem of city modeling from LiDAR point clouds. Roof height estimation and building reconstruction is often the most valuable information to extract from such data, which is acquired using terrestrial or aerial devices. In particular, nadir or near-nadir acquisitions pose a specific constraint, as facades are missed by scanners. Current state-of-the-art approaches have been extensively reviewed (Wang et al. 2018) and can be divided in three categories.

Data-driven methods are probably the most popular techniques. These are bottom-up approaches, in which parametric primitives are extracted from the data and assembled to form a reconstructed model. Existing pipelines typically consist of three successive steps : classification, segmentation and geometric modeling. The semantic interpretation of the data, and the clustering of buildings into individual structures may involve statistical arguments (Poullis and You 2009) or discriminative geometric features coupled with energetic formulations (Lafarge and Mallet 2011). However, the models produced by such techniques do not achieve the same level of detail. For instance, the methods proposed in (Zhou and Neumann 2009) and (Poullis 2013) only reconstruct multi-level flat buildings from airborne point clouds, which is suitable for Manhattan-like districts but less accurate for residential areas. In contrast, the algorithm of (Sohn et al. 2008) considers a binary space partitioning tree to generate LOD2 polyhedral meshes from a point cloud. The one of (Zhou and Neumann 2010) reaches a similar level of detail by minimizing 2.5D quadric error functions, i.e. taking into account both the surface being reconstructed and its projected boundary. However, due to the projection of the points on a 2D grid, the reconstructed facades show a zig-zagging effect, which might be corrected by the mesh simplification procedure described in (Zhou and Neumann 2012). The cell decomposition approach proposed in (Kada and McKinley 2009) allows to reconstruct buildings with a compact representation, but requires precise building footprints as input. The method of (Lafarge and Mallet 2011) also returns persuasive results
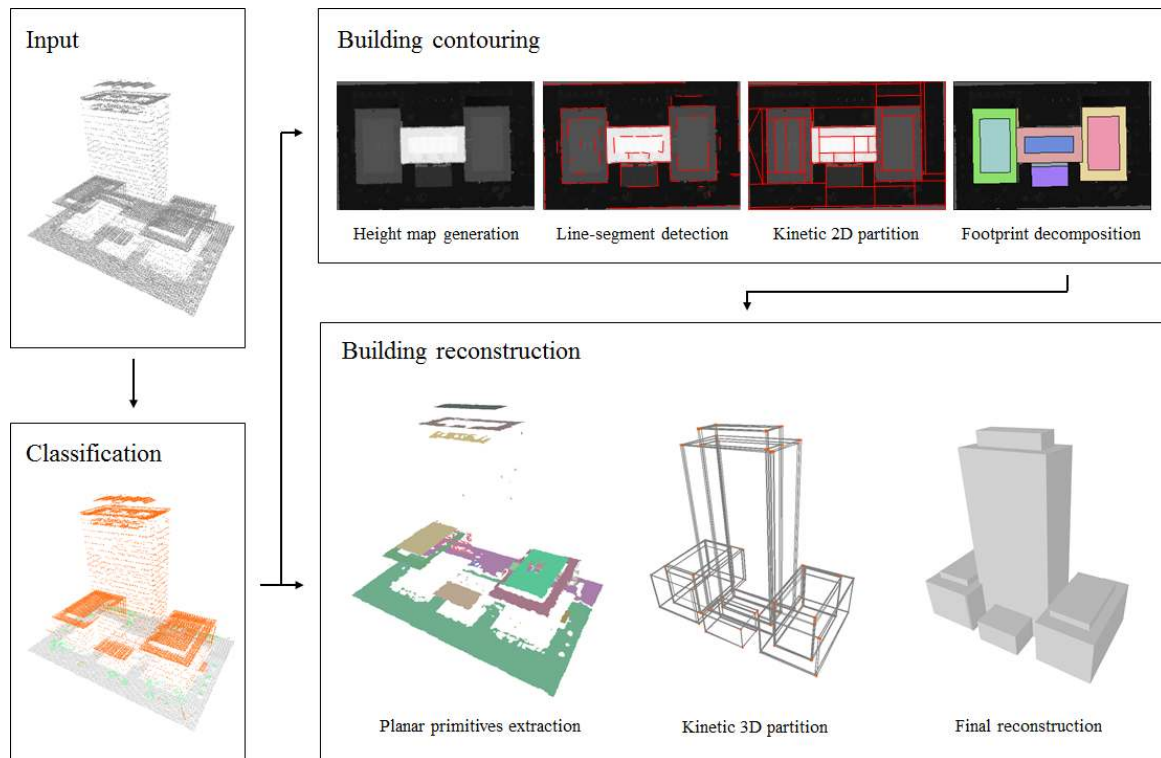
*Corresponding author

Figure 1. Overview of our pipeline. Our method consists of three main steps. We first label points of the LiDAR scan as ground, vegetation or roof. Then, we apply a contouring algorithm to the height map, revealing the facades initially absent in the point set. Finally, we extract and propagate planar primitives from the point cloud, dividing the space into polyhedra that are labelled to obtain a 3D reconstruction of buildings.

over large-scale areas, but suffers from the same failing. Recently, deep-learning-based methods have also been developed in a context of LOD2 urban modeling and achieve very promising results (Zhang and Zhang 2018).

Model-driven methods, for their part, represent the opposite, top-down strategy. This family of techniques considers a predefined library of template structures (e.g. flat, gable, hip or mansard roofs) that is matched to the input data. The work of (Verma et al. 2006) offers a first example of model-driven algorithm : elements of the point cloud are first classified as flat or non-flat and a roof topology graph is considered to decompose a complex building into simpler structures. Another example is the work of (Huang et al. 2013) in which a stochastic approach is used to select the roof templates that best fit the input data. Also requiring building footprints as prior knowledge, the method of (Henn et al. 2013) uses Ransac and supervised machine learning techniques to generate a LOD2 reconstruction of a sparse LiDAR point in a model-driven way. However, such approaches may lack of flexibility with respect to the variety of urban landscapes. Finally, hybrid-driven methods try to take the best of both worlds : parameterized primitives are extracted and assembled with respect to a set of constraints derived from constructive solid geometry (Xiong et al. 2014) (Lafarge et al. 2010).

In the following, we present an algorithm that addresses the problems we exposed before by exploiting powerful computational geometry tools. Given an airborne input point cloud, we design a scalable and data-driven algorithm that generates LOD2 representations of different urban environments as concise polyhedral meshes.

3. ALGORITHM

3.1 Input

As depicted by Figure 1, our algorithm takes as input a point cloud with oriented normals. Normals can be easily estimated thanks to the acquisition system information. If not provided, then basic mathematical tools like principal component analysis can be used.

3.2 Classification

The first step of our pipeline consists in assigning semantic labels to points of the point cloud. Three labels are considered : *ground*, *vegetation*, and *building*.

To this end, we rely on the classification package provided by the CGAL library (Giraudot and Lafarge 2019). For each point of the input dataset, this method computes multi-scale geometric features such as elevation, planarity or vertical dispersion for instance. Extra features provided along with the dataset, like the number of returns, are also taken into account. Given a ground truth training set, these features are then used to train a classifier. The default choice for the classifier is a random forest, that constructs several decision trees to assign each point to one of the three aforementioned subsets.

Interactively labelling the data to get a representative training dataset is a tedious work. However, the CGAL library offers the possibility to save and reuse a trained classifier, which is particularly useful for processing urban scenes of similar nature (dense or surburban areas, downtowns, historical centers...).

3.3 Building contouring

Facades of buildings are often partially or completely missing in aerial point clouds. To obtain accurate reconstructions of cities from such data, we first need a robust method that detects significant height discontinuities in the classified point cloud.

To this end, we project all points on a horizontal, uniformly sampled grid, in order to generate two kinds of maps : (i) a height map, normalized as a grayscale image, (ii) a probability map, measuring the proportion of projected points labelled as *buildings* in each cell of the grid.

These two maps are then processed by the polygonal partitioning technique of (Bauchet and Lafarge 2018). Given an input image, this algorithm first detects line-segments, which are linear approximations of regions where the image gradient is high and regular. These line-segments propagate across the image, until intersecting each other, resulting in a decomposition of the image into convex polygons. Here, we generate a polygonal decomposition of the height map. Intuitively, the detected line-segments, later included in the edges of the partition, correspond to regular height discontinuities in a given direction, i.e. to facades.

Using the probability map previously defined, we further assign a binary activation variable to each polygonal cell, indicating if it is part, or not, of a building footprint. In order to simplify the partition, and decrease the number of cells, we finally apply a clustering algorithm that merges neighbor cells upon condition that there is no height discontinuity at their common border.

3.4 Building reconstruction

To obtain a LOD2 reconstruction from an oriented point cloud, from which we discard all points labelled as *vegetation*, we propose an algorithm in three steps.

First of all, we extract planar primitives from the point cloud. We apply a region-growing algorithm, implemented in the CGAL library (Oesau et al. 2019). A plane hypothesis is iteratively propagated from a point to its neighbors. It is accepted if it has a minimum number of inliers N, with respect to a maximal point-to-plane distance ε. If input points are noisy, more robust methods such as efficient Ransac (Schnabel et al. 2007) or structure-aware shape collapse (Fang et al. 2018) can be considered. The threshold N should be set depending on the density of the cloud, so that N points cover an area of 5 m^2 approximately, whereas ε is typically set to 0.5 m. Once all planes have been extracted, we obtain a set of primitives represented by the planar convex hulls of the different sets of inliers associated to those planes.

The second step of our algorithm consists in performing a kinetic propagation of the primitives in the 3D space. Similarly to the work of (Bauchet and Lafarge 2018), we initialize and process a priority queue to predict and sort intersections between extending primitives. We choose to apply homothetic transformations of ratio $(1 + t)$, where $t \geq 0$ represents the current simulation time. Assuming the existence of a bounding box containing all primitives, processing all intersections results in a polyhedral decomposition of the space.

However, predicting the intersection time between two planar primitives is a costly operation. For large datasets with millions of points, the simultaneous propagation of thousands of primitives is a very complex and time-consuming operation. On the other hand, vertical planes corresponding to facades cannot be extracted from the input point cloud, since the data is missing.

This is the reason why we split the spatial propagation problem into F subproblems, where F is the number of polygonal footprints returned by the procedure described in section 3.3. More precisely, for each footprint we get the list of primitives that intersect or are included in it by projection, and perform a spatial propagation restricted to the dimensions of the footprint. We obtain a set of F 3D subgraphs $G_1, G_2, \ldots G_F$.

The third and final step of our pipeline consists in labelling the polyhedra of each subgraph G_i as inside or outside the buildings to reconstruct. The facets at the interface between outside and inside polyhedra then correspond to the output surface.

We use a voting scheme, based on the observation that in aerial datasets, points delimit the upper parts of the objects of interest. Let P_i be the set of polyhedrons of the subgraph G_i. For each polyhedron $p_j \in P_i$, where $j = 1, 2 \ldots |P_i|$, we initialize a counter c_j to 0. All polyhedrons located below (resp. above) any plane inlier decrement (resp. increment) their counters.

Let us now consider a vector X with $|P_i|$ binary activation variables : $x_j = 1$ (resp. $x_j = 0$) if the polyhedron p_j is labelled as inside (resp. outside) a building to extract. We measure the quality of an output surface using a two-term energy of the form

$$U(X) = D(X) + \lambda V(X) \qquad (1)$$

$D(X)$ is a data term that encourages the selection of a polyhedron p_j when $c_j < 0$, and its rejection when $c_j > 0$. We have :

$$D(X) = \frac{1}{|\mathcal{I}|} \sum_{j=1}^{|P_i|} d_j(x_j) \qquad (2)$$

where

$$d_j(x_j) = \begin{cases} -c_j & \text{when } x_j = 0 \\ c_j & \text{when } x_j = 1 \end{cases} \qquad (3)$$

and $|\mathcal{I}|$ is twice the number of inliers. $V(X)$, for its part, is a generalized Potts model that penalizes the total area of the surface :

$$V(X) = \frac{1}{A} \sum_{j \sim k} a_{jk} \cdot 1_{x_j \neq x_k} \qquad (4)$$

where $j \sim k$ denotes an adjacency relationship between two polyhedra p_j and p_k, a_{jk} is the surface of their common facet, and A is a normalization term defined as the sum of the areas of all facets of the subgraph G_i.

Given a balancing term $\lambda \in [0, 1]$, the optimal surface that minimizes energy U is determined by a min-cut algorithm (Boykov and Kolmogorov 2004). A low value of λ returns a too large and too complex surface, while a high value of λ tends to shrink it. In our experiments, we typically set λ to 0.5.

4. EXPERIMENTS

Datasets. We tested our algorithm on four datasets, representing various urban landscapes. The covered cities are listed in Table 1. The size is given in millions of points.
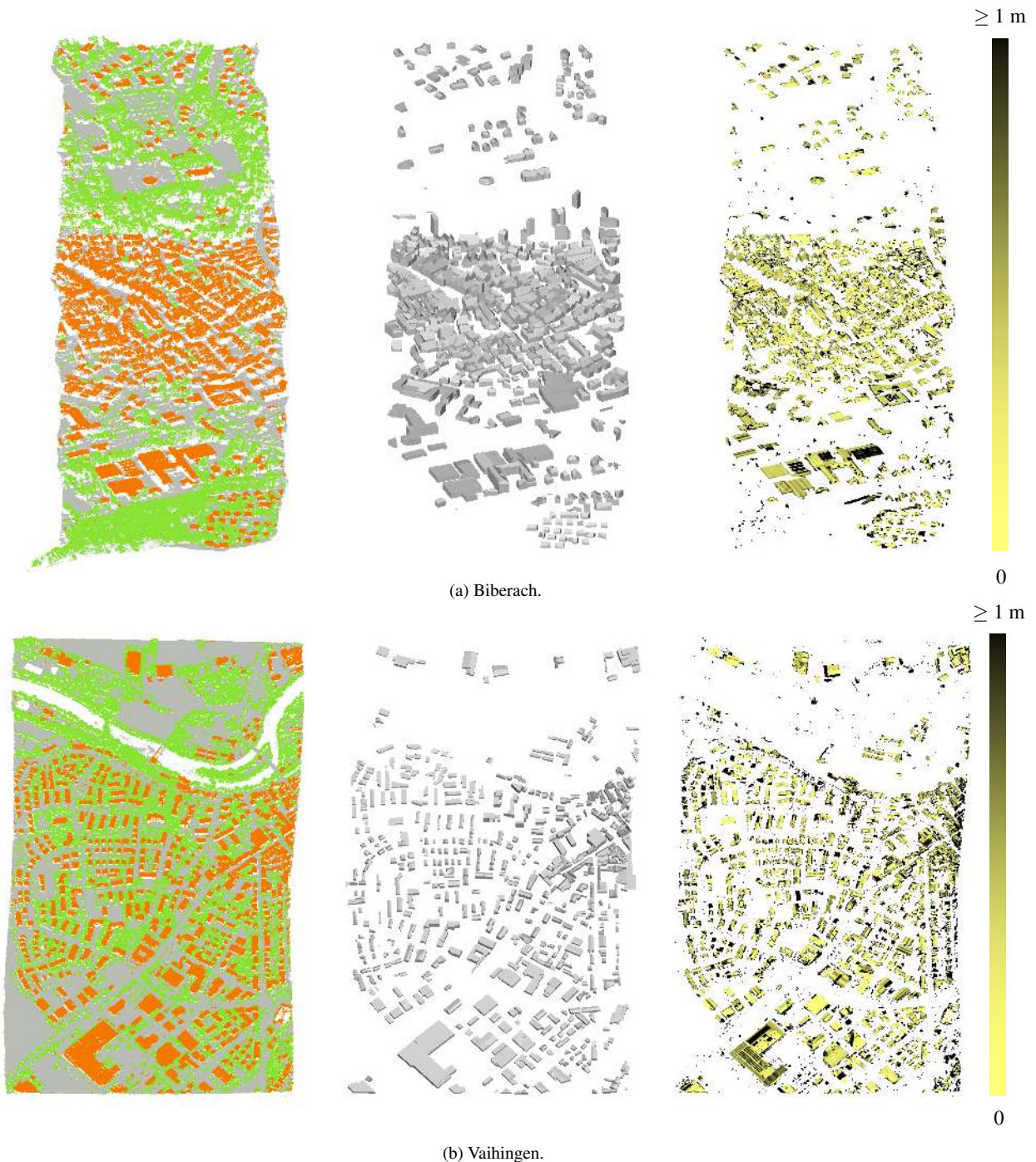
(a) Biberach.

(b) Vaihingen.

Figure 2. Results on European-style urban landscapes. Left column : classified point clouds. Points labelled as ground, vegetation or buildings are colored in gray, green and red, respectively. Center column : reconstructed models. Right column : altimetric error map, in which a darker color represents a larger error. From left to right : classified point cloud, reconstructed model, and altimetric error map. Some close-ups are showed in Figures 4 and 5.

Qualitative results. We present in Figures 2, 3, 4 and 5 the models generated by our algorithm for these cities. From a qualitative point of view, we obtain persuasive LOD2 reconstructions of most buildings. The Biberach and Vaihingen datasets contain a lot of gable and hip roofs which are correctly approximated by our algorithm. As for the Portland and San Diego datasets, our technique also succeeds in determining intermediate levels in complex structures, as well as tilted roofs if any. Facades, which are almost completely missing in the input scans, are generally recovered by a single plane in a given direction. Some of them, however, might be affected by an unwanted zig-zagging effect, reflecting errors in the building contouring procedure. Note that trees could be also reconstructed in 3D by template matching (Verdie and Lafarge 2012) to better represent the urban landscapes.

(a) Portland.

(b) San Diego.

Figure 3. Results on American-style urban landscapes. Some close-ups are showed in Figures 4 and 5.

City	Type	Size	Density (pts/m^2)
Biberach	Historical center	2.3M	3.0
Vaihingen	Residental area	7.3M	6.3
Portland	Downtown	8.7M	7.8
San Diego	Downtown	4.5M	1.6

Table 1. Presentation of the dataset.

Our measures are listed in Table 2.

City	Raw error (m)	Corrected error (m)
Biberach	0.85	0.31
Vaihingen	1.91	0.43
Portland	1.75	0.53
San Diego	2.34	0.62

Table 2. Geometric error for each dataset.

Quantitative results. In Figures 2 and 3, we provide altimetric error maps for each dataset. To generate them, we restrict the input point cloud to elements labelled as rooftops, and compute the one-sided Hausdorff distance from every point to the reconstructed surface. This way, we can evaluate the precision of our method. We compute a *raw* reconstruction error : the mean one-sided Hausdorff distance. However, this measure may be biased. Indeed, isolated mislabelled points in the classification process, and a few buildings missed by the contouring algorithm and further ignored in the reconstruction phase, tend to overestimate the average reconstruction error. That is why we suggest, on an indicative basis, a *corrected* reconstruction error, which discards points located at more than 3 meters from a building which corresponds to the average height of a floor.

The geometric error is typically caused by : (i) undetected superstructures on rooftops, (ii) uncorrectly approximated primitives in the plane extraction procedure (e.g. a unique plane approximates the two sides of a gable roof) and (iii) the resolution of the generated height map, that may shift the extracted footprints towards one direction or another.

Performances. We list in Table 3 the performances of our algorithm for our two biggest datasets, Vaihingen and Portland, in terms of memory peak and running times. Measures were perform on a machine equiped with an Intel® Core™ i7-6700HQ processor clocked at 2.60 GHz and a 32 GB RAM. The obtained values demonstrate the ability of our algorithm to process large volumes of data in a short time.
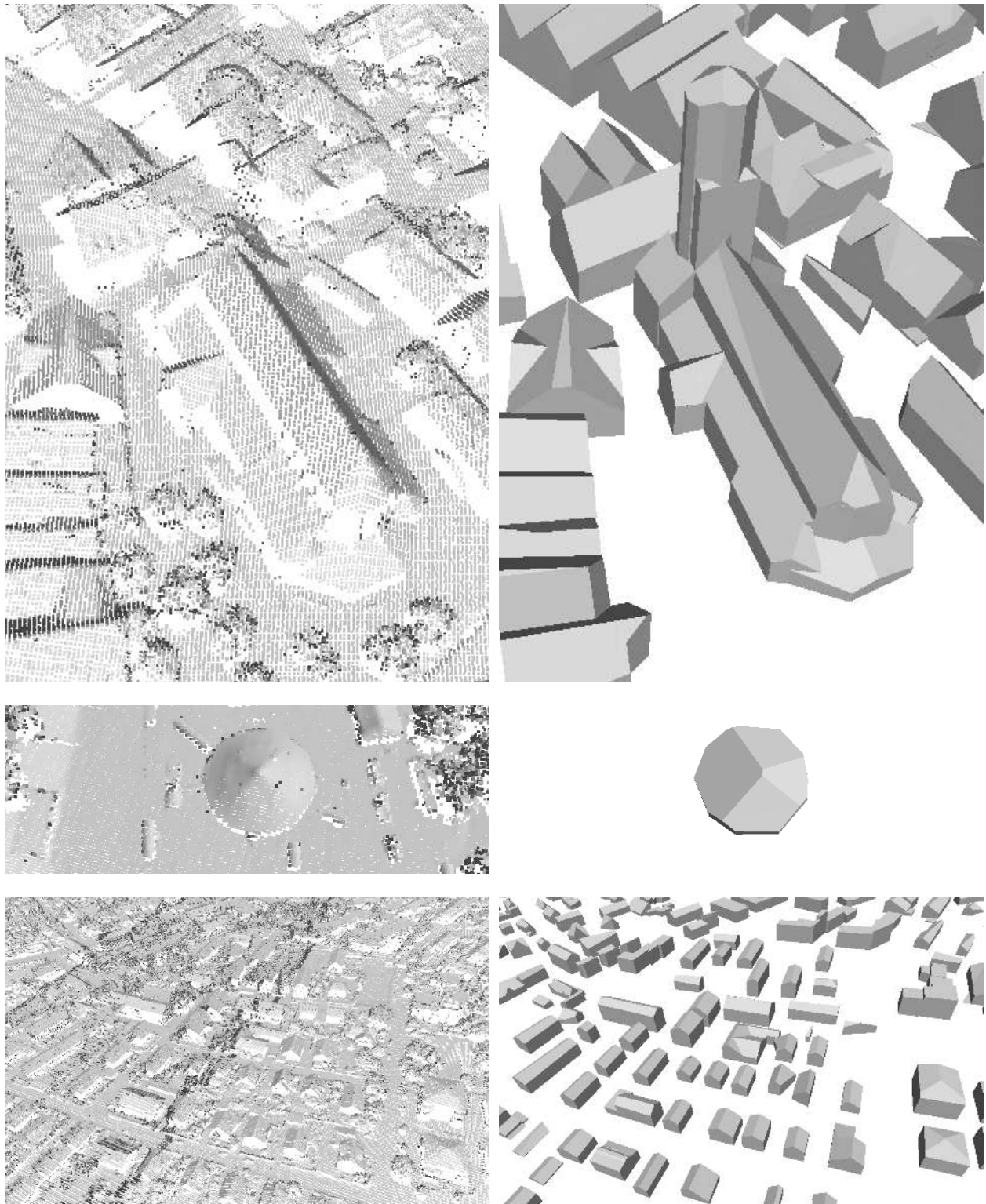
Figure 4. Close-ups on the reconstructed models. From top to bottom : church (Biberach), tent (Biberach), residential area (Vaihingen).
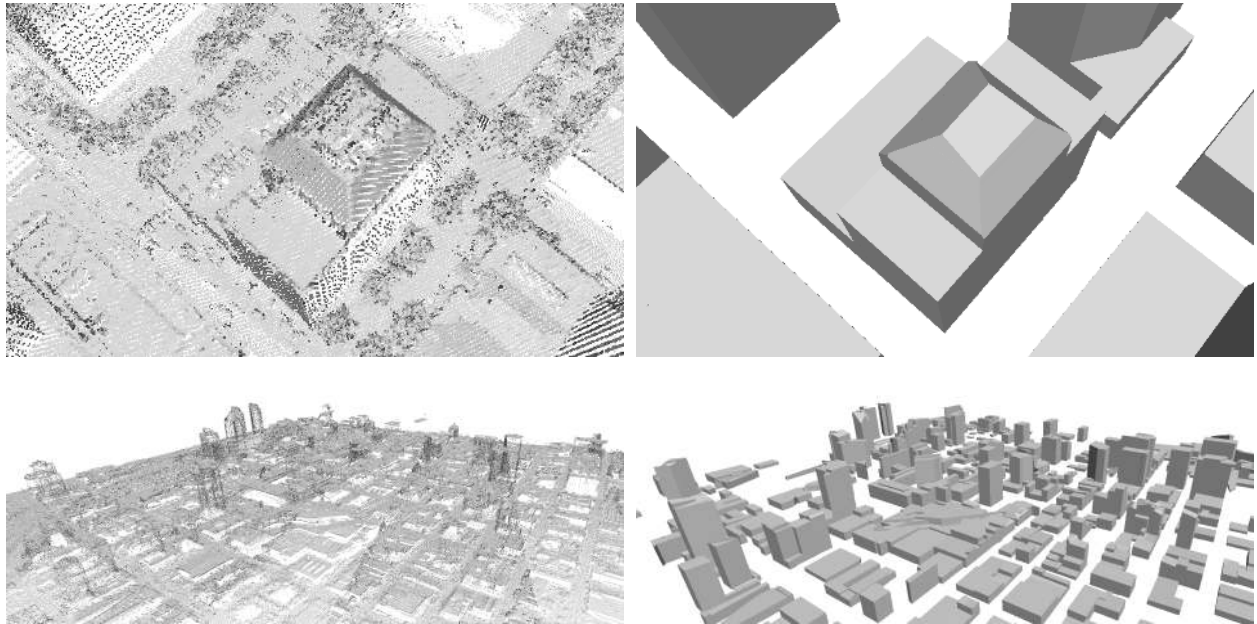
Figure 5. Close-ups on the reconstructed models. From top to bottom : multi-level building (Portland), urban landscape (San Diego).

	Vaihingen	Portland
Building contouring (s)	13.5	20.9
Building reconstruction (s)	509.3	872.8
Memory peak (GB)	2.2	2.6

Table 3. Performance measures for datasets Vaihingen and Portland.

Limitations. Despite its advantages, our algorithm suffers from a few shortcomings. Any error caused by an intermediate step of the pipeline has an impact on the final result. Parts of the cloud representing buildings mislabelled by the classifier will not be reconstructed. Besides, the reconstruction is very sensitive to building contouring errors : if some footprints or intermediate heights inside multi-level structures are missed in the coutouring process, then buildings will also be missing or badly approximated in the final result.

Besides, our primitive detection scheme only extracts planes from the input point cloud. This feature is sufficient for reconstructing accurately most buildings, but free-form shapes like domes or curved walls will only be approximated as a set of planar shapes. Small structures like dormers and chimneys may also be ignored in the reconstructed model, if the minimal number of inliers by the plane extraction procedure is too high.

5. CONCLUSIONS

In this paper, we presented a pipeline for automatically reconstructing a urban scene from an airborne LiDAR scan at the level of detail LOD2. We used a kinetic approach, in which a set of predetected line-segments and planar polygons propagate to decompose the 2D and 3D spaces into cells that are labelled and assembled in our final model. Our approach is fast, scalable and delivers simple polyhedral meshes. It returns promising results on various datasets, representing different types of urban environments.

In future works, we plan to refine the building contouring algorithm in order to simplify and improve the accuracy of our reconstructed models. We might resort to deep-learning-based methods to this end. Another research path would consist in integrating non-linear primitives to our kinetic scheme to achieve better reconstructions of free-form structures.

References

Bauchet, J.-P. and Lafarge, F. (2018), Kippi: Kinetic polygonal partitioning of images, *in* 'Proc. of the Computer Vision and Pattern Recognition (CVPR)'.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S. and Çöltekin, A. (2015), 'Applications of 3D city models: state of the art review', *ISPRS International Journal of Geo-Information* **4**(4).

Boykov, Y. and Kolmogorov, V. (2004), 'An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision', *Trans. on Pattern Analysis and Machine Intelligence* (9).

Duan, L. and Lafarge, F. (2016), Towards large-scale city reconstruction from satellites, *in* 'Proc. of the European Conference on Computer Vision (ECCV)'.

Fang, H., Lafarge, F. and Desbrun, M. (2018), Planar Shape Detection at Structural Scales, *in* 'Proc. of the Computer Vision and Pattern Recognition (CVPR)'.

Giraudot, S. and Lafarge, F. (2019), Classification, *in* 'CGAL User and Reference Manual', 4.14 edn, CGAL Editorial Board.

Gröger, G., Kolbe, T. H., Nagel, C. and Häfele, K.-H. (2012), 'Ogc city geography markup language (citygml) encoding standard'.

Haala, N. and Kada, M. (2010), 'An update on automatic 3d building reconstruction', *ISPRS Journal of Photogrammetry and Remote Sensing* **65**(6).

Henn, A., Gröger, G., Stroh, V. and Plümer, L. (2013), 'Model driven reconstruction of roofs from sparse lidar point clouds', *ISPRS Journal of Photogrammetry and Remote Sensing* **76**.

Huang, H., Brenner, C. and Sester, M. (2013), 'A generative statistical approach to automatic 3d building roof reconstruction from laser scanning data', *ISPRS Journal of Photogrammetry and Remote Sensing* **79**, 29–43.

Kada, M. and McKinley, L. (2009), '3d building reconstruction from lidar based on a cell decomposition approach', *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **38**(Part 3).

Kelly, T., Femiani, J., Wonka, P. and Mitra, N. J. (2017), 'Bigsur: Large-scale structured urban reconstruction', *Trans. on Graphics* **36**(6).

Lafarge, F., Descombes, X., Zerubia, J. and Pierrot-Deseilligny, M. (2010), 'Structural approach for building reconstruction from a single DSM', *Trans. on Pattern Analysis and Machine Intelligence* **32**(1).

Lafarge, F. and Mallet, C. (2011), Building large urban environments from unstructured point data, *in* 'Proc. of the International Conference on Computer Vision (ICCV)'.

Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., Van Gool, L. and Purgathofer, W. (2013), 'A survey of urban reconstruction', *Computer Graphics Forum* **32**(6).

Oesau, S., Verdie, Y., Jamin, C., Alliez, P., Lafarge, F. and Giraudot, S. (2019), Point set shape detection, *in* 'CGAL User and Reference Manual', 4.14 edn, CGAL Editorial Board.

Poullis, C. (2013), 'A framework for automatic modeling from point cloud data', *Trans. on Pattern Analysis and Machine Intelligence* **35**(11).

Poullis, C. and You, S. (2009), Automatic reconstruction of cities from remote sensor data, *in* 'Proc. of Computer Vision and Pattern Recognition (CVPR)'.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J. D., Breitkopf, U. and Jung, J. (2014), 'Results of the isprs benchmark on urban object detection and 3d building reconstruction', *ISPRS Journal of Photogrammetry and Remote Sensing* **93**.

Salinas, D., Lafarge, F. and Alliez, P. (2015), 'Structure-Aware Mesh Decimation', *Computer Graphics Forum* **34**(6).

Schnabel, R., Wahl, R. and Klein, R. (2007), Efficient ransac for point-cloud shape detection, *in* 'Computer Graphics Forum', Vol. 26, Wiley Online Library.

Sohn, G., Huang, X. and Tao, V. (2008), 'Using a binary space partitioning tree for reconstructing polyhedral building models from airborne lidar data', *Photogrammetric Engineering and Remote Sensing* **74**(11).

Verdie, Y. and Lafarge, F. (2012), Efficient Monte Carlo sampler for detecting parametric objects in large scenes, *in* 'Proc. of the European Conference on Computer Vision (ECCV)'.

Verdie, Y., Lafarge, F. and Alliez, P. (2015), 'LOD Generation for Urban Scenes', *Trans. on Graphics* **34**(3).

Verma, V., Kumar, R. and Hsu, S. (2006), 3d building detection and modeling from aerial lidar data, *in* 'Proc. of Computer Vision and Pattern Recognition (CVPR)'.

Wang, R., Peethambaran, J. and Chen, D. (2018), 'Lidar point clouds to 3-d urban models : a review', *Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **11**(2).

Xiong, B., Elberink, S. O. and Vosselman, G. (2014), 'A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds', *ISPRS Journal of Photogrammetry and Remote Sensing* **93**.

Zebedin, L., Bauer, J., Karner, K. and Bischof, H. (2008), Fusion of feature-and area-based information for urban buildings modeling from aerial imagery, *in* 'Proc. of the European Conference on Computer Vision (ECCV)'.

Zeng, H., Wu, J. and Furukawa, Y. (2018), Neural procedural reconstruction for residential buildings, *in* 'Proc. of the European Conference on Computer Vision (ECCV)'.

Zhang, L. and Zhang, L. (2018), 'Deep learning-based classification and reconstruction of residential scenes from large-scale point clouds', *Trans. on Geoscience and Remote Sensing* **56**(4).

Zhou, Q.-Y. and Neumann, U. (2009), A streaming framework for seamless building reconstruction from large-scale aerial lidar data, *in* 'Proc. of Computer Vision and Pattern Recognition (CVPR)'.

Zhou, Q.-Y. and Neumann, U. (2010), $2.5d$ dual contouring: A robust approach to creating building models from aerial lidar point clouds, *in* 'Proc. of the European Conference on Computer Vision (ECCV)'.

Zhou, Q.-Y. and Neumann, U. (2012), 2.5 D building modeling by discovering global regularities, *in* 'Proc. of Computer Vision and Pattern Recognition (CVPR)'.

Zhu, L., Shen, S., Gao, X. and Hu, Z. (2018), Large scale urban scene modeling from mvs meshes, *in* 'Proc. of the European Conference on Computer Vision (ECCV)'.