An Unsupervised Building Footprints Delineation Approach for Large-Scale LiDAR Point Clouds

Xin Xu
xinxu629@umd.edu
University of Maryland at College Park
College Park, Maryland, USA

ABSTRACT

We design a novel unsupervised approach to delineate building footprints on large-scale LiDAR point clouds. By computing an α-shape on low-height points, we delineate the building bottoms on the ground. We then use the terrain ruggedness index and vector ruggedness measurement on the entire points to find flat surface areas. Finally, valid building footprints are filtered by checking flat surfaces in the detected bottom areas. Compared to the Artificial Intelligence (AI)-assisted mapping results from *Microsoft Building Footprints*, the accuracy of the proposed method is 17% higher in the test areas. The simple and effective pipeline makes the proposed method easy to use and suitable for a wider range of applications.

CCS CONCEPTS

• **Computing methodologies** → **Shape analysis**; • **Information systems** → **Geographic information systems**.

KEYWORDS

Unsupervised learning, LiDAR, building footprint, α-shape

ACM Reference Format:
Xin Xu. 2022. An Unsupervised Building Footprints Delineation Approach for Large-Scale LiDAR Point Clouds. In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3557915.3565986

1 INTRODUCTION

This work is presented as part of the 11th SIGSPATIAL Cup competition[1] which considers the problem of delineating building footprints from point clouds generated by airborne laser scanning (ALS). ALS can preserve the 3D structure of objects in large-scale point clouds with high spatial resolution. The possibility of detecting spatial objects from ALS point clouds has been proved in existing studies, such as single tree extraction [10]. Nowadays, ALS point clouds have been explored for automatically generating building maps at a large scale in various applications, such as cellular network planning [2] and smart city planning [1]. To achieve accurate building maps, it is first necessary to identify every building and compute the footprint of detected objects (i.e., building boundary) in the point cloud.

As large-scale ALS point clouds also contain a large portion of vegetation, the pipeline of traditional methods includes distinguishing vegetation points from building points [11], which complicates the entire process. New deep learning techniques have also been utilized in the semantic segmentation from LiDAR data, including building detection [9]. At the same time, the high computational requirements and sophisticated system design raise the bar for users using such tools. Furthermore, external information, like satellite images, is also imported to help delineate buildings in the point clouds [2]. However, such external information is not feasible everywhere and introduces more challenges in registering different data sources. To provide an effective and easy-to-use building footprint delineation method for large-scale point clouds, we propose a novel unsupervised approach that works on points without any other external information. Applying the α-shape [3] of low-height points, we delineate holes on the ground as boundaries of building bottoms. We use the terrain ruggedness index [7] and vector ruggedness measurement [4, 8] to find flat surface areas. The final building footprints are filtered by checking the flat surfaces in the bottom areas. The competition results show that the proposed method achieves better mapping results than *Microsoft Building Footprints*[2] in the experimental areas.

2 METHOD OVERVIEW

Our building footprint delineation method contains three steps. Given input ALS point cloud, we first detect building bottoms on the ground. Then, we identify the building tops from the entire point cloud as plane areas. Finally, we filter valid buildings by finding detected bottoms that cover enough flat areas.

2.1 Detecting building bottoms

We assume buildings are impenetrable and leave no return points at the low level of ALS point clouds. Thus, the first step aims to delineate building bottoms where there are no laser return points at the low height (i.e., ground points). The ground points are already labeled in the competition data. Otherwise, other tools (i.e., lastools [5]) can detect ground points. Then we generate a Digital Terrain Model (DTM) from low points. The cell width is 1m, and the value of each cell is the average elevation value of points inside the cell. Thus, the buildings are located in cells without any value. Then, we collect the center points of empty cells and generate a α-shape [3] with the value of $\alpha = 1.1m$ (see Section 3). Finally, the boundaries of connected components in the α-shape delineate building bottoms.

[1] https://sigspatial2022.sigspatial.org/giscup/index.html

[2] https://www.microsoft.com/en-us/maps/building-footprints

SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9529-8/22/11…$15.00
https://doi.org/10.1145/3557915.3565986

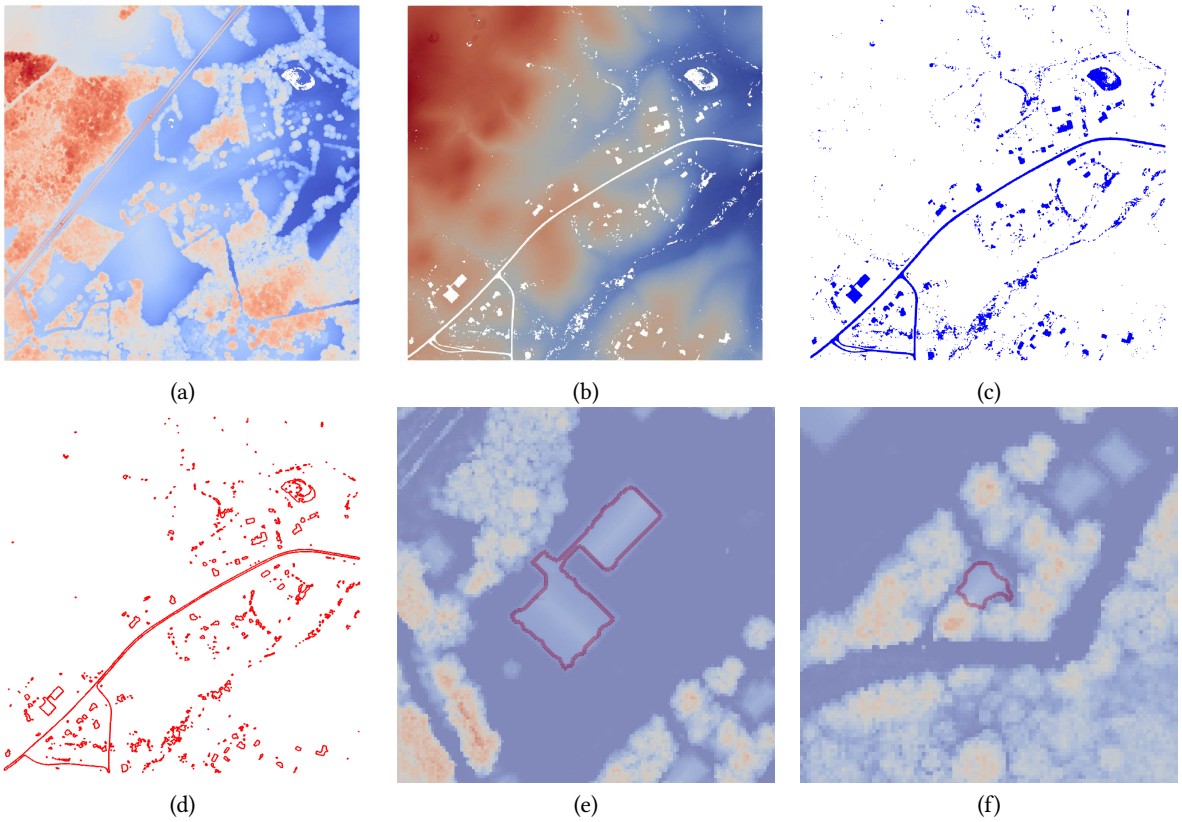

Figure 1: Example of delineated polygons at the ground points. (a) Input point clouds colored by elevation in a blue-red colormap. (b) Generated Digital Terrain Model (DTM) with empty cells colored in white. (c) Center points of empty cells. (d) Generated α-shape of center points. (e) Delineated bottom polygon for a big building. (f) Delineated bottom polygon for a building surrounded by trees.

We present a running example in Figure 1. Input point clouds are colored based on elevation in Figure 1(a). The generated DTM is shown in Figure 1(b), where empty cells, which are also called pits, are presented in white. The center points of such pits are presented in Figure 1(c). We generate an α-shape of pit points that outlines the boundaries of objects in Figure 1(d). Two building details are zoomed in Figure 1, where a large building is well delineated in Figure 1(e). Even surrounded by trees, our method can still detect the building bottom in Figure 1(f).

2.2 Detecting flat surfaces

However, as presented in Figure 1(d), other objects are also detected from low points except buildings. As buildings usually have planar roofs, which are different from other uneven objects (e.g., treetops), our goal in this step is to outline flat surfaces. We fill empty cells in the generated DTM by interpolating the value of an empty cell as the average value of its nearby cells. Then we calculate the height over the DTM for all points and generate a digital surface model (DSM) in a grid with a width of 1m. Working on DSM, we use the terrain ruggedness index (TRI) [7] and vector ruggedness measurement (VRM) [4, 8] to measure the planarity of surface. TRI [7] expresses the height difference between one cell and its adjacent cell. To compute the TRI value of each cell, we first compute the height difference between it and its eight neighbor cells. We then calculate the mean value of squared height differences. The TRI measurement value for the cell is the square root of the calculated mean value. Unlike TRI, VRM [4, 8] measure the surface roughness by considering both slope and aspect. For each cell, we first set a neighbor region NxN (i.e., 3x3). Later, we decompose the slope and aspect into 3-dimensional vectors and calculate the resultant vector magnitude $|V_r|$ within the neighbor region. In the end, the VRM value of the cell is computed as $1 - \frac{|V_r|}{N^2}$.

After calculating TRI and VRM over DSM, we find flat cells if $v \subset [TRI_{min}, TRI_{max}]$, or $v \subset [VRM_{min}, VRM_{max}]$, where v is the cell value, $TRI_{min}, TRI_{max}, VRM_{min}, VRM_{max}$ are threshold values, which are discussed in Section 3. Then, similar to the previous step, we compute the α-shape over flat cells.

Figure 2(a) and (b) shows the TRI and VRM grid generated on points in Figure 1(a), respectively. Flat points are presented in Figure 2(c) and our computed α-shape is showed in Figure 2(d). We observe that flat roof areas from the big building and tree-surrounding building are well delineated(see Figure 2(e) and (f)).

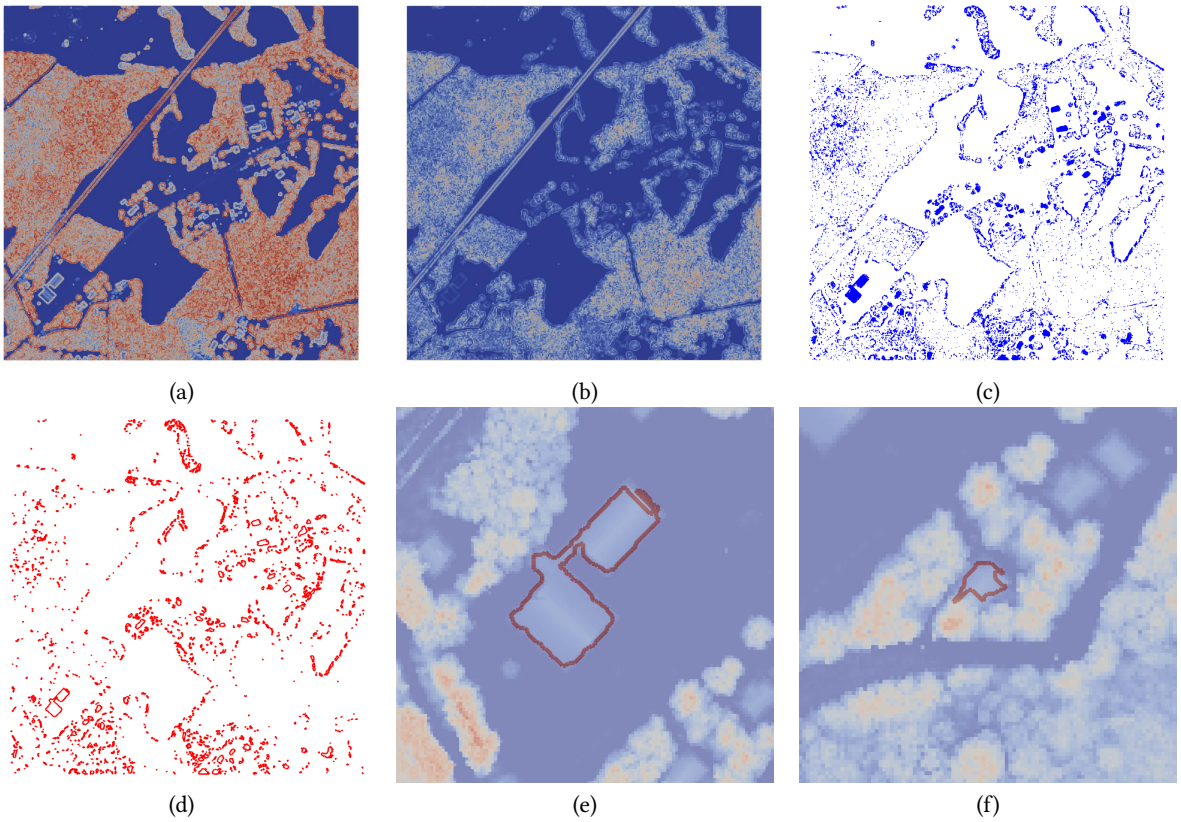

Figure 2: Example of delineated polygons at the top points. (a) Generated terrain ruggedness index (TRI) grid of input points, colored based on the TRI value in a blue-red colormap. (b) Generated vector ruggedness measure (VRM) grid. (c) Center points of selected flat cells. (d) Generated α-shape of center points. (e) Delineated top polygon for a big building. (f) Delineated top polygon for a building surrounded by trees.

2.3 Filtering buildings

In the last step of our method, we filter valid buildings, where the detected building bottoms should cover enough flat surfaces. We provide a series of heuristic filtering to find buildings correctly. First, we consider rectangularity for each detected bottom P_b. The rectangularity (RI) of P_b is computed as $RI = \frac{P_b \cap P_{mb}}{P_b \cup P_{mb}}$, where P_{mb} is the minimum bounded rectangle of P_b. If a large bottom b has a larger RI value than the set value th_{RI}, we mark the bottom as a valid building footprint. Otherwise, we search the flat surface t overlapped with b and calculate the Intersection with Union (IoU) [6] between b and t. If the value of IoU is larger than the set value th_{IoU}, we also mark b as a valid building footprint.

Figure 3(a) shows the final detected buildings. We also compare our delineated building with the satellite images from Google Earth. Large building with the concave shape is well captured (see Figure 3(b)). Even a small building surrounded by trees, our method can still delineate it in Figure 3(c).

3 EXPERIMENTAL RESULTS

Dataset. The 11$^{\text{th}}$ SIGSPATIAL Cup competition includes five test areas in various environments. For example, a town in North Caroline within the humid subtropical climate zone is surrounded by vigorous forests. Three towns within the warm summer continental climate zone are located in Montana, New York, and Wisconsin, respectively. Moreover, a city in New Mexico under the zone of tropical and subtropical steppe climate sits on an unforested, dry and grassy plain. On average, each test area includes 39 buildings in an extensive area (1,592,057 m^2). The LiDAR data is provided by the United States Geological Survey (USGS) as part of the 3D Elevation Program (3DEP)[3]. And, the ground truth building footprints are provided by the competition board.

Results. The validation is based on Intersection over Union [6]. For one test area i, given ALS-derived results O and ground truth G, the IoU value is calculated as $IoU_i = \frac{Area(O) \cap Area(G)}{Area(O) \cup Area(G)}$, where $Area(O)$ is the union of all polygons in O and $Area(G)$ is the union of the all polygons in G. If more polygons are detected in O, IoU_i is further scaled as $IoU_i = IoU_i \cdot \frac{|G|}{|O|}$, where $|*|$ shows the number of polygons. The final accuracy is set as the average of IoU_i for the total five test areas.

[3]https://www.usgs.gov/3d-elevation-program

(a) (b) (c)

Figure 3: (a) Final delineated building footprints. (b) The detected big building over the satellite image. (c) The detected tree-surrounding building over the satellite image. Both satellite images are provided by Google Earth.

Our proposed method can delineate buildings automatically from ALS point clouds with an average IoU value of 0.62, which is 17% higher than the baseline results achieved by Microsoft Buildings. Visually checking our results, evident buildings are correctly outlined (see Figure 3(b)). Buildings adjacent to trees can also be delineated, such as the one in Figure 3(c). Several parameters used in our method reflect the real physical characteristics of buildings, which makes them easy to adjust. During the experiment, with manually tuned parameters, the overall accuracy can be increased 10%. However, the default settings can be starting points and provide good results. Internal key parameter settings are discussed below.

Parameter settings. The key parameters in our method include the value of α, threshold values of terrain ruggedness index (TRI) [7] and vector ruggedness measurement (VRM) [4, 8] to filter flat cells and the values of rectangularity and IoU to find valid buildings.

The value of α controls the α-shape from points, which is used in delineating both building bottoms and flat surfaces (see Section 2.1 and 2.2). We set a fixed value of 1.1m, which is suitable for drawing the boundaries of buildings of interest (i.e., with each width larger than 2.2m). We use TRI and VRM to filter flat cells (see Section 2.2). The smaller value of TRI means that the cell is flatter. We set TRI_{min} as 0, and $TRI_{max} = 0.22m$. Considering the grid width is 1m in our method, the 0.22m also means a low slope (12°). For the VRM value, 0 shows the flat, and 1 means the most rugged. Inspired by the study by Sappington et al. [8], we found that $VRM_{min} = 0$, and $VRM_{max} = 0.05$ are applicable for test areas. Two threshold values rectangularity th_{RI} and th_{IoU} are used to find valid buildings (see Section 2.3). Our method finds building footprints if the delineated shape is very close to a rectangle. Thus we set a high value of $th_{RI} = 0.72$. The IoU value is used to check whether the bottom area has a flat surface, and we set $th_{IoU} = 0.36$.

4 CONCLUSIONS

We design a novel unsupervised approach to delineate building footprints on large-scale LiDAR point clouds. Our approach is effective and easy to use. Moreover, unlike sophisticated deep learning-based methods [9], our approach does not need a time-consuming and data-hungry training process. The heuristic parameter settings in our method are physically meaningful and easy to adjust for better performance. Furthermore, the simplicity of our approach (i.e., no need to distinguish vegetation and buildings in the points) makes it suitable, for example, for interactive applications where the user wants to quickly visualize the result of building delineation on large-scale ALS point clouds.

ACKNOWLEDGMENTS

This work has been supported by the US National Science Foundation under grant number IIS-1910766. We thank Dr. Leila De Floriani and Dr. Federico Iuricich for their help and valuable comments.

REFERENCES

[1] Saleh Albeaik, Mohamad Alrished, Salma Aldawood, Sattam Alsubaiee, and Anas Alfaris. 2017. Virtual cities: 3d urban modeling from low resolution lidar data. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–4.

[2] Philip E Brown, Yaron Kanza, and Velin Kounev. 2019. Height and facet extraction from LiDAR point cloud for automatic creation of 3D building models. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 596–599.

[3] Herbert Edelsbrunner. 2011. Alpha shapes-a survey. In *Tessellations in the Sciences: Virtues, Techniques and Applications of Geometric Tilings*.

[4] RD Hobson. 1972. Surface roughness in topography: quantitative approach. Methuen.

[5] Martin Isenburg. 2022. *LAStools—Efficient tools for LiDAR processing.* https://github.com/LAStools/LAStools

[6] Paul Jaccard. 1908. Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat.* 44 (1908), 223–270.

[7] Shawn J Riley, Stephen D DeGloria, and Robert Elliot. 1999. Index that quantifies topographic heterogeneity. *intermountain Journal of sciences* 5, 1-4 (1999), 23–27.

[8] J Mark Sappington, Kathleen M Longshore, and Daniel B Thompson. 2007. Quantifying landscape ruggedness for animal habitat analysis: a case study using bighorn sheep in the Mojave Desert. *The Journal of wildlife management* 71, 5 (2007), 1419–1426.

[9] Takayuki Shinohara, Haoyi Xiu, and Masashi Matsuoka. 2020. Semantic segmentation for full-waveform LiDAR data using local and hierarchical global feature extraction. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*. 640–650.

[10] Xin Xu, Federico Iuricich, and Leila De Floriani. 2020. A Persistence-Based Approach for Individual Tree Mapping. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*. 191–194.

[11] Qian-Yi Zhou and Ulrich Neumann. 2008. Fast and extensible building modeling from airborne LiDAR data. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. 1–8.