



Article

Critical Points Extraction from Building Façades by Analyzing Gradient Structure Tensor

Dong Chen ¹, Jing Li ^{1,†}, Shaoning Di ^{1,2,†}, Jiju Peethambaran ^{3,*}, Guiqiu Xiang ¹, Lincheng Wan ¹ and Xianghong Li ¹

- ¹ College of Civil Engineering, Nanjing Forestry University, Nanjing 210037, China; chendong@njfu.edu.cn (D.C.); jingli@njfu.edu.cn (J.L.); shaoningdi@njfu.edu.cn (S.D.); guiqiuxiang@njfu.edu.cn (G.X.); linchengwan@njfu.edu.cn (L.W.); lixianghong@njfu.edu.cn (X.L.)
² School of Geosciences and Info Physics, Central South University, Changsha 410083, China
³ Department of Mathematics and Computing Science, Saint Mary's University, Halifax, NS B3P 2M6, Canada
* Correspondence: jiju.peethambaran@smu.ca
† These authors contributed equally to this work.

Abstract: This paper proposes a building façade contouring method from LiDAR (Light Detection and Ranging) scans and photogrammetric point clouds. To this end, we calculate the confidence property at multiple scales for an individual point cloud to measure the point cloud's quality. The confidence property is utilized in the definition of the gradient for each point. We encode the individual point gradient structure tensor, whose eigenvalues reflect the gradient variations in the local neighborhood areas. The critical point clouds representing the building façade and rooftop (if, of course, such rooftops exist) contours are then extracted by jointly analyzing dual-thresholds of the gradient and gradient structure tensor. Based on the requirements of compact representation, the initial obtained critical points are finally downsampled, thereby achieving a tradeoff between the accurate geometry and abstract representation at a reasonable level. Various experiments using representative buildings in Semantic3D benchmark and other ubiquitous point clouds from ALS DublinCity and Dutch AHN3 datasets, MLS TerraMobilita/iQmulus 3D urban analysis benchmark, UAV-based photogrammetric dataset, and GeoSLAM ZEB-HORIZON scans have shown that the proposed method generates building contours that are accurate, lightweight, and robust to ubiquitous point clouds. Two comparison experiments also prove the superiority of the proposed method in terms of topological correctness, geometric accuracy, and representation compactness.

Keywords: critical points; gradient; gradient structure tensor; simplification; building façade; Semantic3D; DublinCity; Dutch AHN3; photogrammetric point clouds; GeoSLAM



Citation: Chen, D.; Li, J.; Di, S.; Peethambaran, J.; Xiang, G.; Wan, L.; Li, X. Critical Points Extraction from Building Façades by Analyzing Gradient Structure Tensor. *Remote Sens.* **2021**, *13*, 3146. <https://doi.org/10.3390/rs13163146>

Academic Editor: Belen Riveiro

Received: 23 June 2021

Accepted: 3 August 2021

Published: 9 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Building contours are the most significant component of the geometric features of a building. As the most basic and critical feature, building contours provide the foundations for scene understanding [1], semantic annotation [2], and 3D abstract perception [3]. In the initial stage of computer vision, a majority of contouring methods are image-based because the “linear” features of images are more accurate when the resolution is guaranteed. Further, the contour of images has a clear definition: the image pixels at the discontinuities in gray level, color, texture, etc. [4]. These discontinuities produce four types of edge profiles which are step, ramp, roof, and ridge edges [5]. The building contour feature is an important feature for image segmentation [6], image understanding [7], and image recognition [8]; however, as image-based contouring cannot fully describe the 3D geometric shapes of complex objects, it is not directly applicable to applications in 3D scene recognition and geometric expression. Recently, the company ESRI (<https://www.esri.com/en-us/home>, last accessed on 6 August 2021) used the deep learning framework, i.e., MaskRCNN [9], to extract accurate building contours from high-resolution aerial and satellite imagery. The

deep learning framework is capable of learning complex semantics and obtaining accurate building contours by training a network on a big dataset. Using deep learning for building contouring from aerial and satellite imagery is also computationally efficient compared with digitizing building features manually; however, it should be noted that the created building contours from aerial and satellite imagery are only 2D building footprints rather than 2.5D building highmaps or 3D building envelopes.

In the past two decades, with the improvement of LiDAR sensor technology, the density and precision of point clouds have been significantly improved. The point clouds are sufficient to describe the global structure features and the local detailed geometric features, which facilitate critical points extraction from point clouds [10]. Despite this, it should be noted that the outdoor scenes are complex and the scans are frequently contaminated by noise, outliers, and missing data caused by occlusion and/or self-occlusion. These adverse factors present a huge challenge for the intelligent extraction of building contours. One should be aware that the term “critical points” has not yet been clearly defined in the context of point clouds [11]. Compared with feature extraction from the images, the feature extraction from the discrete point clouds is far from being mature, and there is still much room for improvement. In this paper, we explicitly define the term “critical points” in the context of 3D point clouds to avoid ambiguity. It contains three types of point sets: (1) corner points: the intersections of three non-parallel planes; (2) edge points: the intersections between two consecutive pairs of planes; (3) boundary points: the outer boundaries of planes, the boundaries of the inner holes, such as the frames of the windows, and the boundaries caused by missing data.

Critical point extraction is the basis of many applications, such as 3D reconstruction [12–15], registration [16–18], target detection [19–21], and data simplification [22]. The critical points of buildings are important of the intermediate inputs for 3D reconstruction. For instance, Mineo et al. [13] propose an algorithm to generate tessellated meshes from point clouds by combing boundary points (a subset of critical points) extraction method and a Fourier transform based spatial filtering. The two stage method of [13] is insensitive to predefined thresholds and superior to methods based on polynomial fitting for 3D reconstruction. Critical points of the objects can also be regarded as the feature points for the registration of multiple scans. For example, Choi et al. [18] use the organized structure information (such as 3D shape information and photometric texture information) of RGB-D images to effectively detect critical points. These detected critical points are applied in edge-based pair-wise registration and a pose-graph SLAM problem based on this registration. The advantages of the registration algorithm based on the detected critical points are verified from both qualitative and quantitative perspectives. Critical point extraction is also the basis of many target detection methods. For example, Wang et al. [21] first extract the critical points of windows, and then realize detection and recognition of window targets by considering the position of windows. This method is robust to noisy data but is not suitable for window detection in complex and large-scale scenes. Data simplification can be achieved by extracting the critical points of objects. In this line, Song and Feng [22] propose a simplification algorithm to reduce the number of points of mechanical parts. This algorithm first identifies these critical points, and then gradually deletes the least important data points until the specified data reduction ratio is reached. The effectiveness of the simplification [22] is verified by the simplified results of several actual point cloud datasets.

The contour feature of point clouds is fundamental to many applications. We can generally categorize the existing contouring methods into two types based on the spatial dimensions in which the contouring is performed: image-based contouring methods and LiDAR-based contouring methods.

(1) Image-based Contouring: The contouring methods in images are more mature than in 3D point clouds. Many methods project the input 3D point cloud onto an image, and directly use the existing contouring methods of images to extract contour pixels. In this line, some methods convert point clouds into grayscale images. For example, Li et al. [23] generate a corresponding grayscale image and then determine the critical points based

on the contours detected in the grayscale image; however, this method is only for the extraction of the roof contours of buildings. Similarly, Poullis [24] converts the building roof points into a binary image, and then use the contouring method in the binary image to extract the roof contours. Converting the point clouds into a distance image is another commonly used method. Wang et al. [25] detect the contours in the 2D grayscale image and the distance image. Then multiple sets of contour pixels generated from grayscale and distance images are combined to finalize the contour generation from 3D point clouds. This method not only extracts the contours of the buildings but also regularizes it. Apart from the above methods, there are methods that establish the correspondences between the image pixels and the corresponding 3D point clouds to maintain accurate feature pairs. For example, Li et al. [26] use the elevation difference in the point clouds to extract rough contours, and then project these rough contours onto the image for compact contour extraction. This method has strong robustness in complex scenes. Chen et al. [27] match a single scene of point clouds with a single corresponding image to realize critical point detection. This method reduces the computational complexity.

(2) LiDAR-based Contouring: Since the 3D point clouds contain the intrinsic structure information and topological relationship of object parts, the intrinsic information of 3D point clouds, such as curvature, normal vector, and gradient can also be used for the extraction of critical points. For instance, Yang and Zang [28] use local curvature as an edge index and select points with curvature higher than the given threshold as critical points. This method is suitable for contouring in small objects with high-quality point clouds. Demarsin et al. [29] first perform first-order segmentation and extraction of critical points based on normal vector estimation, and then organize the generated segments into a graph to restore sharp feature lines. Essentially, most of the edges can also be seen as intersections of surfaces. Borges et al. [30] divide the point clouds into planes and then locate critical points by calculating the intersection lines between adjacent planes; however, this method is not suitable for extracting the critical points from the objects with the small fragment of the planes. To solve this problem, Ni et al. [31] project adjacent points on a local plane and then extract critical points according to the angular gap metric. This method can achieve good results from the small scenes with high-quality point clouds, but for large and complex scenes, parameter tuning is an extremely difficult task.

In summary, image-based contouring methods need to rasterize 3D point clouds onto regular 2D images and use more mature image edge detection algorithms to extract various contour pixels; however, this type of method ignores the spatial information and topological relationship of the 3D point clouds and loses precision during coordinate transformation. Although LiDAR-based contouring methods overcome these drawbacks, this type of method is only suitable for contouring some specific objects, such as buildings, road curbs, and tree skeletons. It requires accurate segmentation and target recognition from 3D raw points [32], and it is difficult to control the thickness of the extracted critical points.

In this paper, we propose a method to extract critical points of buildings from 3D discrete point clouds. More specifically, the point confidence indicating the local quality of point clouds is estimated. After that, the 3D gradient of individual point is defined. The gradient of point clouds is successively encoded into 3×3 structure tensor whose eigenvalues represent the distribution of gradients in the local neighborhood. The problem of the façade critical points extraction can be transformed into the problem of analyzing eigenvalues of structure tensor along three eigenvectors. Considering the detected critical points are relatively thick, these critical points are refined by the concept of contour simplification. The proposed method is not only more accurate than the existing methods which directly derive contour features in 3D point clouds, but also superior to the projected contouring methods, which convert 3D point clouds into 2D images followed by image-based contouring [33]. In addition, the results of the proposed method have high precision and the method is capable of controlling the degree of contour thickness easily. Considering our work is built on the previous works, we explicitly denote the novel contributions of the paper.

- Building façade contouring framework: We propose a generic framework for building façade contouring from LiDAR and photogrammetric point clouds. The framework consists of five steps including confidence measure estimation, 3D gradient calculation, gradient structure tensor encoding, dual-threshold criterion, and simplification. These steps loosely coupled interact in the pipeline (see Figure 1) to enhance the flexibility of the framework, thereby achieving a tradeoff between geometric accuracy and compact abstraction of building façades.
- Gradient structure tensor encoding: We encode each point's structure tensor, which describes the gradient variation in the local neighborhood areas. Through analyzing each point's structure tensor, building point clouds can be roughly labeled into corner points, edge points, boundary points, and constant points (see Section 2.3).
- The solid experiments and effective comparisons: We provide qualitative and quantitative performance evaluations using five datasets, and give two comparisons with the state-of-the-art methods to demonstrate the superiority of the proposed method in terms of topological correctness, geometric accuracy, and compact abstraction.

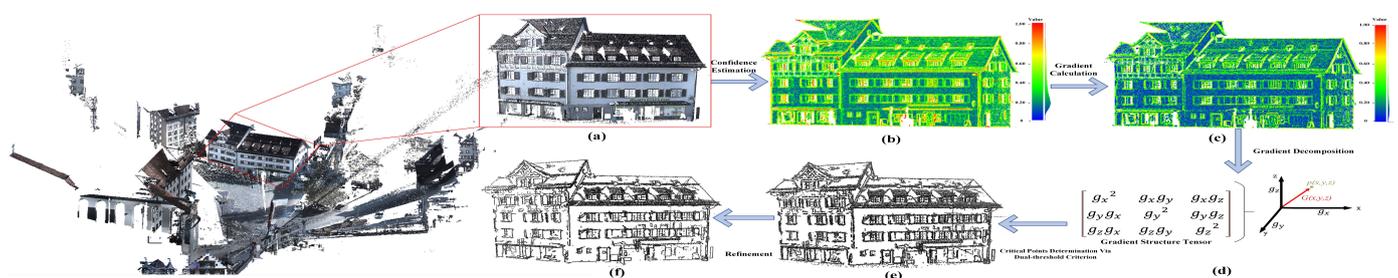


Figure 1. Pipeline of the proposed methodology. (a) The representative building is selected by the manual segmentation. (b) The visualization of confidence measure of building point clouds. (c) Gradient visualization of building point clouds. (d) Gradient structure tensor encoding. (e) The critical point clouds extraction via dual-threshold criterion. (f) The refinement of critical point clouds through simplification.

The reminder of this paper is organized as follows. Section 2 describes the detailed methodology including 3D gradient definition, gradient structure tensor encoding, dual-threshold criterion, and the concept of refinement. In Section 3, the experimental datasets, the performance evaluation of building contours are presented, analyzed, and discussed. Finally, Section 4 concludes the paper along with a few suggestions for future research topics.

2. Methodology

To generate a compact set of critical points of building façades and rooftops (if, of course, such rooftops exist), we propose the methodology that consists of the following steps that are denoted in Figure 1. The confidence measure of each point is estimated by the eigenvalue analysis of covariance matrix established in a local neighborhood (see Figure 1b), followed by the calculation of the gradient of each point cloud in 3D space (see Figure 1c). The point's gradient is then decomposed into three components along with the corresponding coordinate axis. We encode gradient information of the point cloud into a 3×3 structure tensor, whose eigenvalues can reflect the gradient distribution in the local neighborhood of each point. (see Figure 1d). After that, the confidence measure and the gradient structure tensor of each point are both provided as inputs to the proposed dual-threshold method, which determines whether the current point under processing is a critical point or not (see Figure 1e). In the final step, we simplify the results to provide a highly compact set of critical points, thereby enhancing the flexibility of building façade abstraction (see Figure 1f).

2.1. Confidence Estimation

Confidence measures the local quality of point clouds within a small sphere centered at the individual point. Since the purpose of this paper is to extract critical points from building façades, the confidence definition should be directly related to the characteristics of these critical points. Based on this principle, we estimate the confidence by a combination of two geometric properties, i.e., fitting quality C_f and sampling uniformity C_s . The first geometric property represents the fitting quality of the local tangent plane at point p_i , and it can be calculated by: $C_f^i = \lambda_0^i / (\lambda_0^i + \lambda_1^i + \lambda_2^i)$, where $\lambda_0^i \leq \lambda_1^i \leq \lambda_2^i$ represents three eigenvalues of the covariance matrix of point p_i and its local neighborhood points. If point p_i and its local neighbors can perfectly fit the local tangent plane, the value of fitting quality measure C_f^i of p_i approaches 0. In contrast, if the neighbors of point p_i is inhomogeneous, they are most probably distributed on façade edges, façade corners, and façade extrusions and intrusions. In this case, the value of C_f^i of p_i tends to be 1. The second geometric property C_s represents the local sampling uniformity, and it can be calculated by the following equation: $C_s^i = \lambda_1^i / \lambda_2^i$. If point p_i and its local neighbors are distributed linearly, the value of C_s of p_i approaches 0; if uniformly distributed, C_s tends to be 1; therefore, this measure is effective detection of outer boundaries of façades and the window frames, where the density variances and data missing frequently occur due to laser beam penetration through window glass. In addition, to make the calculation of confidence measure more robust, the above two measures are calculated at multiple scales by varying the size of the neighborhood spheres. In our case, we set three scales of the local neighborhood sphere with a radius of 1.0, 1.5, and 2.0 times of mean point density of point clouds. Inspired by the work in [34], the complete confidence measure $C^i \in [0, 1]$ of point p_i is given as below:

$$C^i = 1 - \frac{1}{n} \sum_{j=1}^n (1 - 3C_f^i) \cdot C_s^i \quad (1)$$

where, parameter n represents the number of static scales for individual point's confidence estimation. It is obvious that if the confidence measure C^i of point p_i tends to be 1, it means p_i mostly comes from façade boundaries, corners, windows frames, and other shapes that varied dramatically, as demonstrated in Figure 2. In contrast, if this value approaches 0, the point p_i has high local fitting quality, and it most likely comes from inner points of façades, as evident in Figure 2.

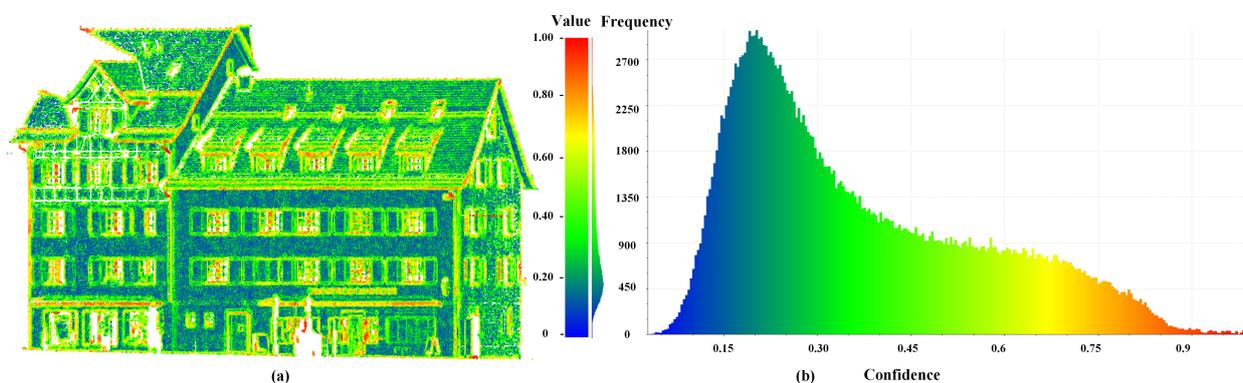


Figure 2. Confidence measure visualization for a representative building. (a) Confidence measure visualization for building façades and rooftops. (b) The histogram distribution of confidence.

2.2. Gradient Definition in 3D Point Cloud Space

Once the confidence measures of point clouds are estimated, we can define the gradient for the discrete point clouds in 3D space based on the confidence measure. We first analyze the directional derivative, from which the gradient can be inferred. The directional

derivative denotes a rate of change of a function in any given direction. The gradient indicates the direction of the greatest change of a function of more than one variable. The magnitude of the gradient is the largest value of all directional derivatives of the current point, and the gradient direction is the corresponding direction of the directional derivative. Let function $f(x, y, z)$ be a differentiable function with three variables and $u = \cos(\alpha)i + \cos(\beta)j + \cos(\gamma)k$ be a direction vector in the 3D space. The directional derivative of function $f(x, y, z)$ along the direction vector u is given by the equation: $D_u f = \lim_{t \rightarrow 0} \frac{f(x+t\cos(\alpha), y+t\cos(\beta), z+t\cos(\gamma)) - f(x, y, z)}{t}$ if the limit exists. Three angles α , β , and γ represent the angles between the directional vector u and positive of three axes. The symbol t represents the distance between the current point and its neighbors. The maximum value of the directional derivative occurs when the gradient $\nabla f(x, y, z)$ and the direction vector u are in the same direction.

In the context of discrete point clouds, the function $f(x, y, z)$ denotes the confidence value at point $p(x, y, z)$. The gradient value $G(x, y, z)$ of point $p(x, y, z)$ can be obtained by maximizing the directional derivative $D_u f$, which is approximately equal to $\max_{j \in N} (\frac{C^i - C^j}{d_{ij}})$, where C^i and C^j represent the confidence measures of current point p_i and its neighborhood point p_j selected from p_i 's neighborhood set N ; d_{ij} denotes the Euclidean distance from point p_i to point p_j . The relationship between gradient and directional derivative in the discrete form of 3D space is given below:

$$G(x, y, z) = \max(D_u f) \approx \max_{j \in N} \left(\frac{C^i - C^j}{d_{ij}} \right). \quad (2)$$

The gradients of individual point cloud are vividly shown in Figure 3. The value of gradient $G(x, y, z)$ of point p_i can be decomposed into three components along three axes:

$$\begin{cases} g_x \approx \max_{j \in N} \left(\frac{C^i - C^j}{d_{ij}} \right) \times \frac{d_{ij}^x}{d_{ij}} \\ g_y \approx \max_{j \in N} \left(\frac{C^i - C^j}{d_{ij}} \right) \times \frac{d_{ij}^y}{d_{ij}} \\ g_z \approx \max_{j \in N} \left(\frac{C^i - C^j}{d_{ij}} \right) \times \frac{d_{ij}^z}{d_{ij}} \end{cases} \quad (3)$$

where d_{ij}^x , d_{ij}^y , and d_{ij}^z represent the coordinate differences between p_i and p_j in x -, y -, and z -axis.

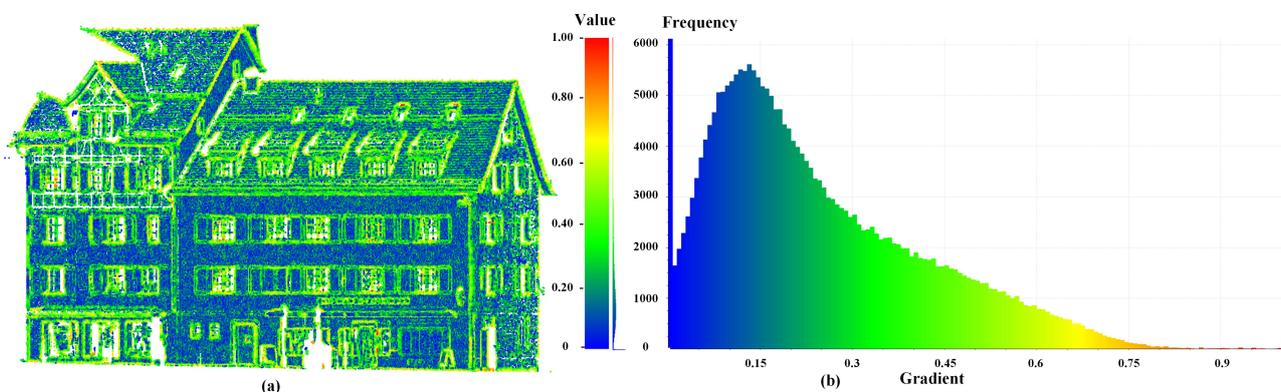


Figure 3. Gradient property visualization for a representative building. (a) Gradient property visualization of building façades and rooftops. (b) The histogram distribution of each point's gradient. Note that gradient values are normalized into the range $[0, 1]$ for better visualization.

2.3. Gradient Structure Tensor Generation

Inspired by Harris's corner detector [35] and edge detector [36] using the intensity changes by shifting the windows in a small amount in various directions, we calculate the changes of the point gradient by shifting a small amount in various directions $(\Delta x, \Delta y, \Delta z)$ in 3D point cloud space. The change of gradient $E(\Delta x, \Delta y, \Delta z)$ of a point cloud $p(x, y, z)$ in the local neighborhood set N can be represented as below:

$$E(\Delta x, \Delta y, \Delta z) = \sum_{p(x,y,z) \in N} [G_{x+\Delta x, y+\Delta y, z+\Delta z} - G_{x,y,z}]^2 \quad (4)$$

Equation (4) can be transformed into Equation (5) using Taylor expansion with $O(\Delta x^2 + \Delta y^2 + \Delta z^2)$ as the reminder term:

$$E(\Delta x, \Delta y, \Delta z) = \sum_{p(x,y,z) \in N} [\Delta x \cdot g_x + \Delta y \cdot g_y + \Delta z \cdot g_z + O(\Delta x^2 + \Delta y^2 + \Delta z^2)]^2 \approx (\Delta x, \Delta y, \Delta z) M (\Delta x, \Delta y, \Delta z)^T \quad (5)$$

where $M = \sum_{p(x,y,z) \in N} \begin{bmatrix} g_x^2 & g_x g_y & g_x g_z \\ g_y g_x & g_y^2 & g_y g_z \\ g_z g_x & g_z g_y & g_z^2 \end{bmatrix}$, and it represents the gradient structure tensor

of the local neighborhood point set N centered at current point p_i . M is a semi-positive and symmetric matrix, whose three eigenvalues λ_0^M , λ_1^M and λ_2^M are mutually orthogonal. The gradient distribution in the local neighborhood set N can be estimated by analyzing three eigenvalues of gradient structure tensor M . The current point p_i has four status to be considered:

- Corner points: the current point p_i is most probably at the intersection area of three mutually nonparallel surfaces (façades and rooftop planes). In this case, all three eigenvalues are large.
- Edge points: the current point p_i most likely belongs to the intersection edges generated from façades and/or rooftop planes. In this situation, two eigenvalues are relatively large.
- Boundary points: the current point p_i most probably comes from the outer boundaries or boundaries of inner holes (e.g., window frames) caused by missing data of the façades. In this case, only one large eigenvalue can be observed.
- Constant points: the local neighborhood areas of current point p_i maintain approximately constant gradient values, i.e., arbitrary shifts of 3d voxel windows centered at p_i cause little change value in E (see Equation (4)). All three eigenvalues are small in this case.

Obviously, the term critical points in our paper refer to three types of point sets, including corner points, intersection points, and boundary points. These three parts constitute the skeleton of building façades.

As the gradient matrix, M is calculated as a discrete form rather than continuous representation, the calculated eigenvalues are not stable and robust. That is to say, the obtained gradient structure tensor M cannot fully reflect the gradient distributions in the local neighborhood areas. To solve this deficiency, the gradient structure tensor M needs to be smoothed using the Gaussian weight function, and the smoothing processing is given as follows:

$$\bar{M}_i = \sum_{p_j \in N} \text{Gauss}_{(p_i, p_j)} \cdot M_i \otimes M_j \quad (6)$$

$$\text{s.t. } \text{Gauss}_{(p_i, p_j)} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{\|p_i - p_j\|^2}{2\sigma^2}}$$

where, “ \cdot ” represents the multiplication of numbers and matrices, and the symbol “ \otimes ” represents convolution operation between two gradient structure tensors M_i and M_j produced

by two point p_i and p_j according to Equation (5). \overline{M}_i is the Gaussian smoothed version of M_i . σ is the standard deviation of distance between current p_i and its neighborhood point p_j .

2.4. Dual-Threshold Criterion

As the large or small eigenvalue is hard to be defined, inspired by [35], we directly define the response function using the three eigenvalues of gradient structure tensor M to detect the critical points from building façades. The response function of critical points is defined below:

$$R_{\overline{M}} = \det(\overline{M}) - k[\text{trace}(\overline{M})]^3. \quad (7)$$

$$\text{s.t.} \quad \begin{cases} \det(\overline{M}) = \lambda_0^{\overline{M}} \cdot \lambda_1^{\overline{M}} \cdot \lambda_2^{\overline{M}} \\ \text{trace}(\overline{M}) = \lambda_0^{\overline{M}} + \lambda_1^{\overline{M}} + \lambda_2^{\overline{M}} \end{cases}$$

The value of $R_{\overline{M}}$ expresses the the status of current point by an elegant combination of three eigenvalues of gradient structure tensor \overline{M} in Equation (7). This means that if the value of $R_{\overline{M}}$ is greater than or equal to a predefined threshold $T_{\overline{M}}$, the corresponding point is regarded as the critical point; however, setting an inappropriate threshold $T_{\overline{M}}$ will cause the under- and/or over-segmentation of a façade's critical points. This is most probably because the calculated value of $R_{\overline{M}}$ is a local variable, and it is determined through multiple steps including confidence estimation, gradient tensor generation, and tensor smoothing, using varied scales of local neighborhood sizes. In contrast, $T_{\overline{M}}$ is a global threshold defined in the whole building façade data space, causing the fixed threshold to be unable to fully represent each point's gradient variations estimated at different local neighborhood areas.

As shown in Figure 4, the result with green color shows the extracted critical points using the single-threshold criterion, i.e., determination of critical points by checking whether the condition $R_{\overline{M}} \geq T_{\overline{M}}$ is met. We can see that the building skeleton is relatively thick and some outliers distributed at the rooftops are mistakenly classified as critical points. To alleviate this problem, we employ a dual-threshold criterion by introducing another constraint, i.e., $G(x, y, z) \geq T_G$ where T_G is a predefined gradient threshold. As mentioned previously, the gradient value of each discrete point is calculated by Equation (2). That is to say, if a point cloud is labeled as a critical point, it must simultaneously satisfy the following two conditions: $G(x, y, z) \geq T_G$ and $R_{\overline{M}} \geq T_{\overline{M}}$. As shown in Figure 4, the point clouds with red color are critical points generated using dual-threshold criterion by setting 2.25 and 25 as threshold values regarding T_G and $T_{\overline{M}}$. Compared with results by single-threshold criterion, the obtained critical points using dual-threshold criterion is relatively thin and more reasonable to describe the skeleton of building façades and rooftops. Moreover, some scattered outliers on the rooftop are eliminated, making the results clean and concise.

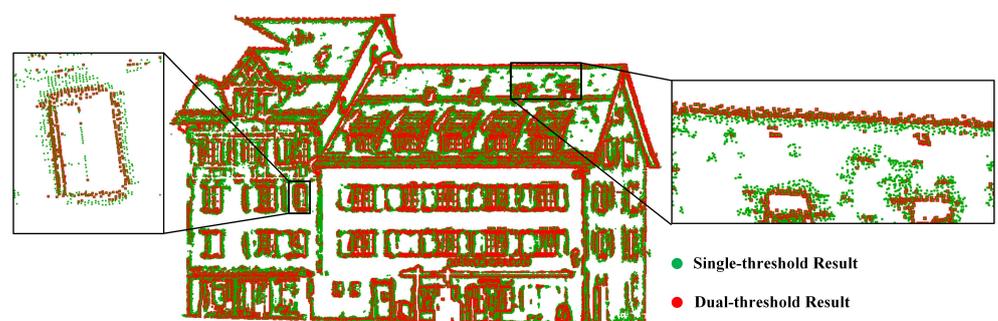


Figure 4. Critical point cloud comparisons by single- and dual-threshold criteria.

2.5. Critical Point Refinement through Concept of Simplification

Although we can also obtain an optimal critical point set by jointly tuning two thresholds, i.e., $T_{\overline{M}}$ and T_G to guarantee the geometric accuracy and high compactness of building

façade representation, it should be noticed that it is not a trivial work to simultaneously tune two parameters. It needs trial and error experiments, leading to time-consuming and labor-intensive work. In this sense, we prefer to obtain a relative thick façade skeleton with thick points on edges and refine the result in the successive simplification step.

To this end, we transform the problem of building skeleton abstraction into the problem of simplification. This is to say, we further refine the dual-threshold result by the concept of simplification to enhance the flexibility of building façade abstraction. To this end, we employ three classic algorithms, i.e., grid simplification, hierarchical simplification [37], and the weighted locally optimal projection (WLOP) algorithm [38]. The grid simplification algorithm divides the space of point clouds into multiple supervoxels, from which only the representative one is chosen to express the corresponding supervoxel. The hierarchical simplification provides an adaptive simplification of the points set through recursively splitting the point clouds using binary space partitioning. Compared with two other simplification algorithms, the WLOP algorithm not only simplifies but also regularizes the critical façades point clouds. WLOP algorithm is an improved LOP algorithm [39] with a weighted density term, which can denoise and remove outliers from imperfect point data and produce an evenly distributed set of particles that faithfully adheres to the captured shapes [38]. The consolidated points generated by WLOP are newly created instead of being chosen from the critical point set. These three refinement results with different inputs are shown in Figure 5.

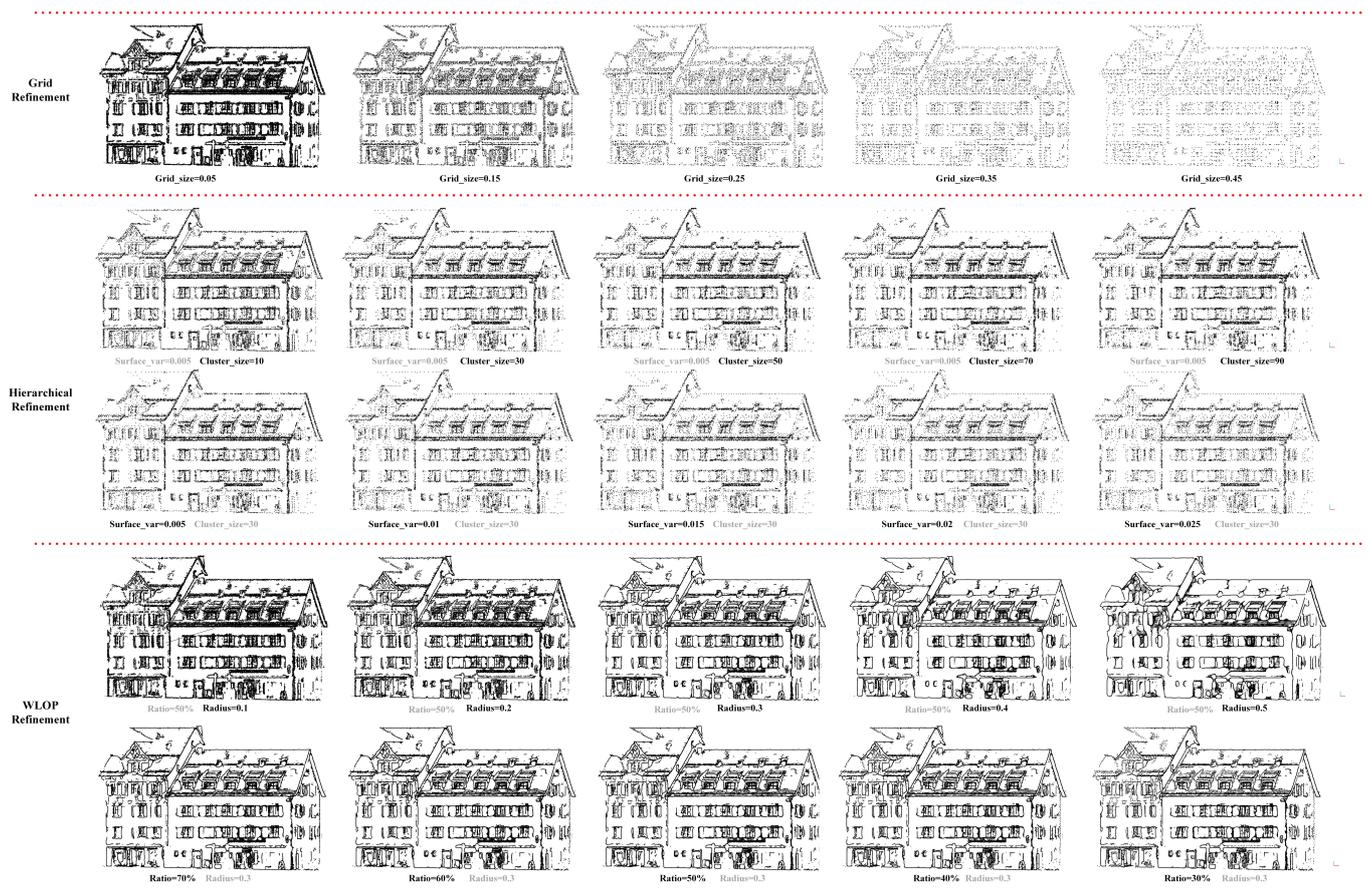


Figure 5. The refinement results generated by the grid, hierarchical, and WLOP algorithms with different input parameters.

In fact, there is no “absolute criterion” to determine whether the created critical point clouds are optimal or not. The generated critical points with fewer point clouds tend to be more lightweight/compact. The critical point clouds with high compactness are suitable for storage, web transmission, acceleration of rendering large-scale scenes, and other VR and/or AR applications. Although having compactness has such strengths, it should be

aware that the geometric accuracy might be weakened if imposing a strict abstraction for building façade point clouds. In this sense, we think an optimal abstraction should strike a balance between geometric accuracy and compactness of critical points. To increase the flexibility of façade representation, we hope the created points are generated at different LoDs rather than a single fixed-level representation.

3. Performance Evaluation

3.1. Dataset Specification

The first dataset we used for experiment is Semantic3D (<http://www.semantic3d.net/>, last accessed on 6 August 2021) [40]. It provides a large-scale outdoor scene of 30 scans over 4 billion labeled point clouds with diverse outdoor scenes. We selected eight representative buildings to verify the critical extraction algorithm. The reasons for the selection of the buildings from Semantic 3D are threefold: (1) Accurate labeled point clouds: each point cloud from Semantic3D is manually assigned a specific class label. We can easily select diverse buildings according to the building label to verify the proposed algorithm. (2) Diverse architectures: all the released scenes are captured in Central Europe, from which diverse European architectures are provided. (3) Inhomogeneous scan points: this dataset is particularly challenging because the scans are acquired using a surveying-grade terrestrial laser scanner (TLS) with a long measurement range, thereby resulting in extreme point density changes and occlusions. Because of this, the buildings are well suited for testing the proposed algorithm. Regarding the detailed descriptions with Semantic3D, we suggest the reader refers to the work in [40].

To evaluate the proposed algorithm's capability for processing building façades with extremely sparse point clouds, we select two groups of buildings from Dutch AHN3 (Actueel Hoogtebestand Nederland) and DublinCity (https://geo.nyu.edu/catalog/nyu_2451_38684, last accessed on 6 August 2021) datasets. The Dutch AHN3 point cloud is acquired by an aircraft during 2014 and 2019. For the acquisition, Riegl LMS-Q680i laser scanning sensor is the most frequently used and sometimes adoption of Riegl VQ-780i [41]. The mean density of point clouds is around 16 points/m². The AHN3 dataset is released to the public through a central distributed platform PDOK (<https://downloads.pdok.nl/ahn3-downloadpage/>, last accessed on 6 August 2021). The DublinCity dataset is acquired by the helicopter in March 2015. The entire dataset consists of 14 flight paths, covering the major areas of the Dublin city center with around 5.6 km². The adopted sensor is a TopEye system S/N 443. The entire dataset consists of 500 × 500 rectangular tiles. The registered point clouds by multiple paths offer the mean point density of 250 to 348 points/m² for different titles. The reader can refer to the work [42] for more details of the DublinCity dataset. These two ALS (Airborne Laser Scanning, ALS) datasets have a clear semantic building label, which facilitates us to choose the buildings with a variety of geometric shapes. The building façades contained by these two ALS datasets are extremely sparse and include missing data due to the building self-occlusions. In addition, the selected two groups have varied sizes and rooftop shapes/structures in different orientations. These two datasets provide an opportunity to test the proposed algorithm's abstraction capability, focusing not only on building façades but rooftops.

To evaluate the algorithm's extensibility, we select one zone MLS (Mobile Laser Scanning, MLS) point clouds from the TerraMobilita/iQmulus 3D urban analysis benchmark [43] (<http://data.ign.fr/benchmarks/UrbanAnalysis/>, last accessed on 6 August 2021). The entire benchmark contains 11 zones with 300 million point clouds in the center of Paris, France. The dataset is acquired in January 2013 by Stereopolis II, an MLS system developed at the French National Mapping Agency. Two Riegl LMS-Q120i and one HDL-64E Velodyne LiDAR sensors are integrated into Stereopolis II. Our selected zone contains a fully annotated street section 200 m long with 12 million point clouds. We believe this section of MLS point clouds from a dense urban environment in Paris can effectively test the algorithm's extensibility to a large scale.

We also use UAV-based photogrammetric points and hand-held scanning (HLS) points to verify the robustness of the proposed algorithm. As shown in Figure 6, point clouds for Buildings \mathcal{A} and \mathcal{B} at Nanjing Forestry University are derived by the overlap images captured by DJI Phantom 4 RTK with a GSD (Ground Sampling Distance, GSD) of 2.74 cm at 100 meters flight altitude. The front and side overlap rates are set to 80% and 70%, respectively. The acquired oblique images are fed into Bentley ContextCapture commercial software package for generating high-density point clouds. Because of the high density, we downsample the raw point clouds to 3 cm. Another Building \mathcal{C} shown in Figure 7 at Nanjing Forestry University is scanned by GeoSLAM ZEB-HORIZON Scanner, which achieves 3D point cloud registration and stitching in real time based on SLAM (Simultaneous Localization and Mapping) technique. These two types of point clouds have low point precision and high outliers and noise, thereby posing extreme challenges for critical point extraction.

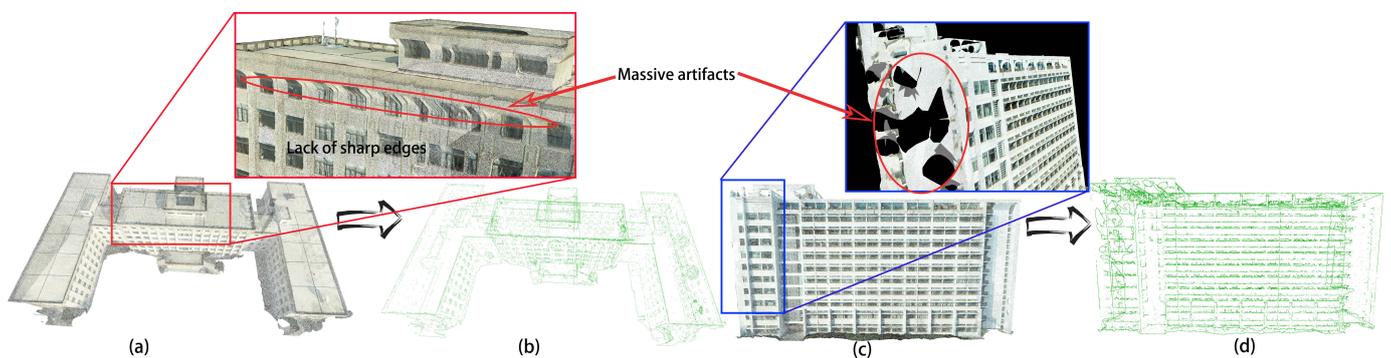


Figure 6. Critical point extraction from UAV-based photogrammetric point clouds. (a,c) are the colored point clouds of two representative Buildings A and B at Nanjing Forestry University. The generated critical points for these two buildings are shown in (b,d). Note that the enlarged rectangle views show that the UAV point clouds miss sharp edge features and have massive artifacts.

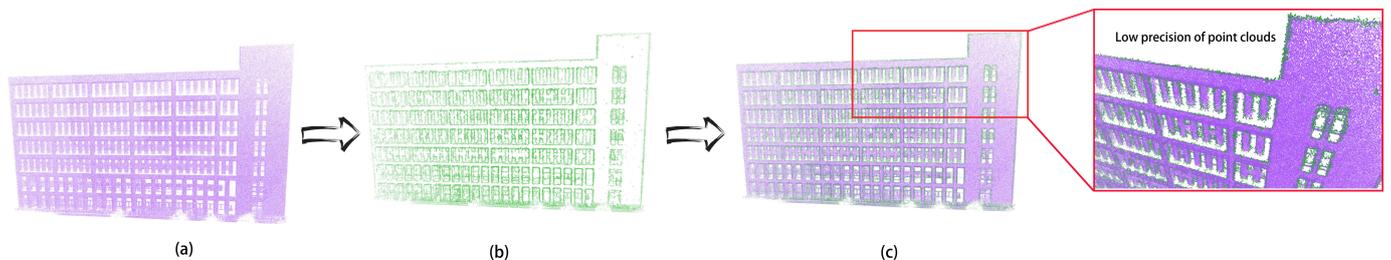


Figure 7. Critical points derived from GeoSLAM ZEB-HORIZON point clouds. (a–c) are the raw points, the façade contour result, and their overlap map for better visualization.

3.2. Parameter Analyzing

Before we evaluate the quality of the created critical points, we first analyze all the relevant thresholds and explain how to select their appropriate values. All the thresholds that are used in the experiments are listed in Table 1. For the confidence estimation, we average the confidence measure at three scales of neighborhood spheres centered on current point p_i with the radius of 1.0, 1.5, and 2.0 times of mean point density (ρ) of a given dataset. Compared with the calculation at a single, fixed scale, our multi-scale strategy reduces parameter insensitivity and enhances the stability and reliability of the confidence measure; however, it should be noted that increasing the size of the radius at a multi-scale will increase the computational complexity. Through trial-and-error experiments, it has been found that the neighborhood sphere $r = \{\rho, 1.5\rho, 2\rho\}$ can strike a balance between the highly reliable confidence calculation and low cost computation. In the steps of gradient calculation, structure tensor generation, and structure tensor smoothing, the neighborhood N of the current point p_i is commonly used. Fortunately, the threshold

N is insensitive to a wide range of values. Through experiments, it has been found that, by selecting in the range [20, 80], the obtained results have no obvious differences. In our case, we simply set $N = 40$ for the sake of achieving a balance between the result stability and the computational efficiency. For the dual-threshold setting, the gradient obeys a normal distribution, i.e., $G(x, y, z) \sim N(\mu, \sigma^2)$, where, μ and σ are the mean and the standard derivation of the gradient that can be obtained by the histogram analysis of a set of individual building points. Threshold T_G is encouraged to be set with a relatively smaller value and it is suggested to be within the range $[0.6\mu, \mu]$. Threshold $T_{\bar{M}}$ controls the strictness of determining whether a point is a critical point or not. Selecting smaller or larger values will cause over- or under-segmentation. Fortunately, this threshold is less sensitive to a wide range of values. It is recommended to be set in the range [5, 30]. These principles can be demonstrated in Figure 8 with varying thresholds T_G and $T_{\bar{M}}$. The grid simplification algorithm has only one input parameter, i.e., cell size ($Grid_size$) of the supervoxel. The larger the cell size is, the lower the geometric accuracy it produces, and less critical points might be generated and vice versa. Generally, we set $Grid_size$ as two or three times the mean point density to meet the requirements of simplification. The hierarchical simplification algorithm has two input parameters, including max cluster size ($Cluster_size$) and max surface variation ($Surface_var$). The larger they are, the fewer sampled points we have. In our case, they are set to 30 and 0.01. The WLOP algorithm also needs two input parameters, including the percentage of points to retain ($Ratio$) and the neighbor size ($Radius$). The former controls the percentage of the obtained critical point clouds, while the latter decides the degree of regularization. More specifically, when we adopt a large neighbor size, we tend to generate the regularized result. In our case, they are simply set to 50% and 0.2 to achieve a certain degree of simplification. More results of the simplification threshold settings are shown in Figure 5. It should be noted that there was no optimal threshold setting for simplification, instead one should consider the geometric accuracy and the degree of abstraction, which are generally determined by practical applications.

Table 1. Parameters for the proposed critical point extraction algorithm. The symbol “-” denotes that the corresponding value can be calculated by the given dataset. “ ρ ” is the mean density of the dataset and “ μ ” is the mean value of the gradient, which can be obtained by the histogram analysis of a set of individual building points.

Modules	Parameters	Recommended Values	Descriptions
Confidence Estimation	r	$(\rho, 1.5\rho, 2\rho)$	Parameter r defines the neighborhood sphere radius for the calculation of confidence measure. Three scales of the spheres are used to obtain the mean value of each point’s confidence.
Gradient Calculation/Structure Tensor Generation/Gaussian Smoothing	N	40	It represents how many points are included for calculating gradient and structure tensor of point p_i .
Dual-threshold Criterion	T_G	$[0.6\mu, \mu]$	Gradient threshold.
	$T_{\bar{M}}$	[5, 30]	Response function threshold.
Gaussian Smoothing	σ	-	The standard deviation of distance between the current point p_i and its neighbor point set p_j ($p_j \in N$).
Grid Refinement	$Grid_size$	$2\rho/3\rho$	It defines the cell size of the 3D grid. The larger cell size is, the less critical points are remained.
Hierarchical Refinement	$Surface_var$	0.01	It controls the local variation of the divided clusters. The values goes from 0 with a coplanar cluster to 1/3 with a fully isotropic cluster. The large it is, the fewer critical points we have.
	$Cluster_size$	30	It controls the maximum number of the divided clusters. The larger it is, the few critical points we have.
WLOP Refinement	$Ratio$	50%	It determines the percentage of points to retain.
	$Radius$	0.2	It controls the degree of regularization. The larger neighbor size is, the more regularized results are obtained.



Figure 8. The critical point extraction from building façades using dual-threshold criterion. It can be seen that the results are slightly different based on the composition of T_G and T_M .

3.3. Compactness

Compactness is a critical index for the evaluation of whether the extracted critical point set is lightweight. That is, if the result contains a few critical points, it represents more compactness of the building façade abstraction. Critical points with high compactness facilitate web transmission, storage-saving, and large-scale building point clouds rendering. We use compactness ratio C_a calculated by dividing the number of critical points by the number of raw building point clouds to represent the degree of compactness. In addition, we also use another compactness ratio C_b calculated by dividing the number of created point clouds by the number of reference point clouds. The reference point clouds are generated by extensive manual work using the CloudCompare open-source tool (<https://www.danielgm.net/cc/>, last accessed on 6 August 2021).

Quantitative evaluation results of the selected buildings from different datasets are listed in Table 2. We can see that for eight Semantic3D buildings, the mean compactness ratio of dual-threshold results approaches 36.96%. In this case, the number of critical points is consistent with the number of reference points, as demonstrated by C_b in the dual-threshold column. This ratio can be further compressed to less than 20% or even 10% according to the different degrees of abstraction via three types of refinement algorithms. We note that high compactness (e.g., 0.92% for Semantic Building 8 after grid refinement) does not mean that the results are more reasonable. Although having a high compactness ratio, the detailed shape features might be ignored on some occasions, thereby weakening abstraction accuracy. For the two other ALS and MLS datasets, the compactness ratio C_a is significantly lower than the TLS dataset (e.g., only 0.72 for DublinCity after hierarchy refinement). This is because the ALS point clouds do not have enough capability to describe the building shape's details due to sparse point density. As UAV and HLS datasets include sufficient shape details due to high-density point clouds, the abstraction accuracy can be guaranteed when posing heavy abstraction (less than 1% after grid or hierarchy refinements). In summary, the compactness is closely related to scene complexity, point cloud density, and a degree of abstraction. We should strike a balance between the compactness and accuracy while guaranteeing an abstraction of critical points.

Table 2. Quantitative evaluation of extracted critical points' compactness. #Building and #Reference represent the number of point clouds for building raw data and reference data. C_a and C_b denote the compactness ratios of the generated critical point clouds with respect to the input buildings and the reference point clouds. Two types of ratios are calculated in two phases, including the dual-threshold judgment stage and the refinement stage using three different simplification algorithms. The symbol "--" means the corresponding data are not available.

Datasets	Building ID	#Building	#Reference	Dual-Threshold		Grid		Hierarchy		WLOP	
				$C_a(\%)$	$C_b(\%)$	$C_a(\%)$	$C_b(\%)$	$C_a(\%)$	$C_b(\%)$	$C_a(\%)$	$C_b(\%)$
TLS	Semantic3D Building 1	78,223	31,971	43.89	107.38	12.88	31.51	8.73	21.37	21.37	52.29
	Semantic3D Building 2	142,346	48,949	37.90	110.21	14.16	41.17	9.00	26.18	18.67	54.31
	Semantic3D Building 3	800,048	258,267	33.00	102.22	2.12	6.54	9.43	29.21	16.50	51.11
	Semantic3D Building 4	713,975	225,596	33.84	107.09	3.99	12.63	8.02	25.39	16.89	53.44
	Semantic3D Building 5	406,525	118,602	37.95	130.09	4.33	14.83	8.50	29.13	18.91	64.83
	Semantic3D Building 6	446,521	101,665	29.65	130.24	0.98	4.32	6.75	29.65	14.83	65.11
	Semantic3D Building 7	169,399	96,377	54.43	95.67	8.25	14.50	11.53	20.27	26.99	47.43
	Semantic3D Building 8	763,570	162,936	25.03	117.31	0.92	4.31	5.42	25.40	12.51	58.63
ALS	Netherland	6,047,084	-	23.11	-	9.08	-	1.15	-	11.55	-
	Dublin	9,189,507	-	14.38	-	6.20	-	0.72	-	7.19	-
MLS	Paris	3,738,321	-	9.61	-	1.23	-	1.46	-	14.81	-
UAV	Building A	11,192,178	-	3.71	-	0.49	-	0.19	-	1.85	-
	Building B	6,475,114	-	6.45	-	0.63	-	0.32	-	3.22	-
HLS	Building C	5,118,751	-	5.79	-	0.56	-	0.32	-	2.89	-

3.4. Accuracy

As previously mentioned, we have the reference of the critical point cloud for each building in the Semantic3D dataset. The reference is generated through manual labeling using the CloudCompare open-source tool. Based on the reference, we evaluate the quality of results generated by the dual-threshold analysis and the successive refinement using grid, hierarchy, and WLOP algorithms. The quantitative evaluation statistics are shown in Table 3. The *Max* values for Buildings 4 and 5 are greater than three meters because some critical points are contaminated by outliers generated by the laser beam penetration through façade window glass, as demonstrated in the areas labeled in black ovals in Figure 9d. Another reason is that our results contain some pseudo-edges/boundaries that are generated in some regions of the building façades corrupted by the missing data due to occlusions, as evident in the region labeled by black rectangles in Figure 9. In fact, these pseudo-edge/boundary points in reference are not included, thereby causing relatively large *Max* values of these two buildings. We also observe that *Max* values for dual-threshold are constantly greater than grid and hierarchy results because these two results are downsampled from the initial dual-threshold result. Although WLOP is also generated from the initial dual-threshold result, the *Max* value of Building 7 is 1.3606 m, which is greater than the dual-threshold result of 1.3306 m. The possible reason could be that the WLOP result is not simply downsampled from the initial dual-threshold critical point set but is a newly regenerated point set through the optimized technique. Apart from the hierarchy *Mean* values of 0.0271 m and 0.0442 m for Buildings 2 and 4, the dual-threshold *Mean* values are less than the *Mean* values in the three refinement algorithms. This means that the refinement through the concept of simplification can weaken the geometric accuracy. This point keeps consistent with the conclusion derived in Section 3.3. The *RMSE* measures the differences between the generated critical point set and the reference. To evaluate the importance of this difference and make the difference independent of the unit, we use another relevant measure $RMSE'$ calculated by dividing the value of *RMSE* to the diagonal length of the corresponding building's bounding box. As we can see, the maximum of $RMSE'$ is 0.0088 for Building 6, which means that the maximum gap between our result and reference is only 0.88% of the building's diagonal.

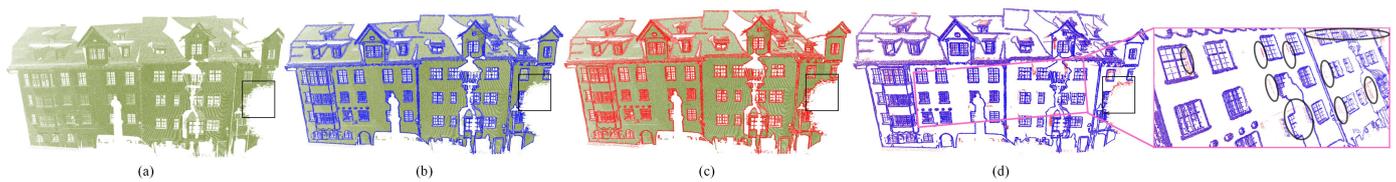


Figure 9. Comparison between dual-threshold result and the corresponding reference data for Semantic3D Building 4. (a) is the raw data of Semantic3D Building 4. (b,c) show that the reference data denoted by blue color and the dual-threshold result denoted by red color are superimposed onto their raw data. (d) is the overlap between the reference and the dual-threshold result.

Table 3. The accuracy statistics for eight representative buildings in Semantic3D dataset. The symbols *Min*, *Max*, *Mean*, and *SD* represent the minimum, maximum, mean, and standard deviation of the Euclidean distance from the critical point to its closest point in the reference. '*RMSE*' measures the *RMS* distance from the extracted critical point to its closest point in the reference. These values are normalized (*RMSE'*) to the diagonal length of the building's bounding box. The first five measures are all in meters, while the last measure *RMSE'* is dimensionless. Four rows of values associated with each building represent the statistics generated by dual-threshold analysis, grid refinement, hierarchy refinement, and WLOP refinement.

Building ID	Max (m)	Min (m)	Mean (m)	SD (m)	RMSE (m)	RMSE'
Semantic3D Building 1	2.1129	0	0.0368	0.1819	0.1856	0.0064
	1.7406	0	0.0415	0.1758	0.1807	0.0063
	1.6855	0	0.0354	0.1632	0.1670	0.0058
	1.7434	0	0.0587	0.1663	0.1764	0.0061
Semantic3D Building 2	2.0195	0	0.0291	0.1137	0.1174	0.0026
	2.0195	0	0.0372	0.1393	0.1442	0.0031
	2.0195	0	0.0271	0.1061	0.1095	0.0024
	1.9955	0	0.0538	0.1080	0.1206	0.0026
Semantic3D Building 3	1.9446	0	0.0052	0.0683	0.0686	0.0017
	1.8483	0	0.0179	0.1010	0.1025	0.0026
	1.9141	0	0.0098	0.0707	0.0713	0.0018
	1.9094	0	0.0203	0.0684	0.0720	0.0018
Semantic3D Building 4	3.7826	0	0.0556	0.2779	0.2838	0.0062
	3.7787	0	0.0936	0.3573	0.3693	0.0081
	3.7118	0	0.0442	0.2283	0.2325	0.0051
	3.7817	0	0.0736	0.2718	0.2816	0.0062
Semantic3D Building 5	3.2625	0	0.0788	0.2582	0.2703	0.0060
	3.2364	0	0.1477	0.3543	0.3839	0.0085
	3.2245	0	0.0769	0.2352	0.2474	0.0055
	3.2474	0	0.0931	0.2479	0.2648	0.0059
Semantic3D Building 6	0.7303	0	0.0568	0.1545	0.1647	0.0087
	0.7226	0	0.0682	0.1525	0.1670	0.0088
	0.7230	0	0.0599	0.1526	0.1639	0.0086
	0.7175	0	0.0699	0.1497	0.1652	0.0087
Semantic3D Building 7	1.3306	0	0.0340	0.1069	0.1122	0.0023
	1.2979	0	0.0559	0.1245	0.1365	0.0028
	1.2749	0	0.0342	0.0998	0.1055	0.0021
	1.3606	0	0.0520	0.1011	0.1137	0.0023
Semantic3D Building 8	1.9282	0	0.0252	0.0948	0.0982	0.0040
	1.8951	0	0.0433	0.1139	0.1218	0.0050
	1.9181	0	0.0279	0.0853	0.0898	0.0037
	1.8918	0	0.0381	0.0923	0.0998	0.0041

3.5. Comparison

We compare the proposed algorithm with two other classic algorithms, including Canny edge detector [33] and Xia's method [36]. As the Canny edge detector is designed for 2D images rather than 3D point clouds, we project the building façades onto 2D gray images according to the dominant façade's normal and location. The value of each pixel is an average gradient value of all point clouds within each pixel. The pixel value is normalized to the range from 0 to 255. Compared with the Canny edge detector that worked on raster images, Xia's and our methods are implemented onto 3D point clouds. The comparison results with these two methods are shown in Figure 10. We can see that our method can effectively extract the critical points from façades with a high degree of complex shapes. In Semantic3D Building 3, the enlarged rectangles denoted by the purple color have a very complex geometric shape with a high degree of nonlinear structure. Despite this, we can obtain a reasonable skeleton for this complex shape; however, some edge points are missing in Xia's result. As our method works directly on the 3D point clouds like Xia's method, this means these two methods can effectively maintain the geometric shapes of building façades without artifacts; however, for the Canny method, it inevitably brings distortions during the conversion from 3D point clouds to 2D grayscale images. This is demonstrated in Building 5 denoted by red rectangles. In addition, it should be noted that the Canny detector often produces an adverse effect of "double edge", which means one edge is represented by two contours due to the thickness/width of the edge, as evident in blue rectangles in Semantic3D Building 5. We should also notice that our method is also insensitive to the density of point clouds. In the enlarged areas of green rectangles in Semantic3D Building 7, the point cloud is extremely sparse and with irregular distribution. In this case, Canny and Xia's results are very messy; however, our method is superior to these two methods because of the reasonable definition of 3D gradient and an accurate analysis of the gradient structure tensor.

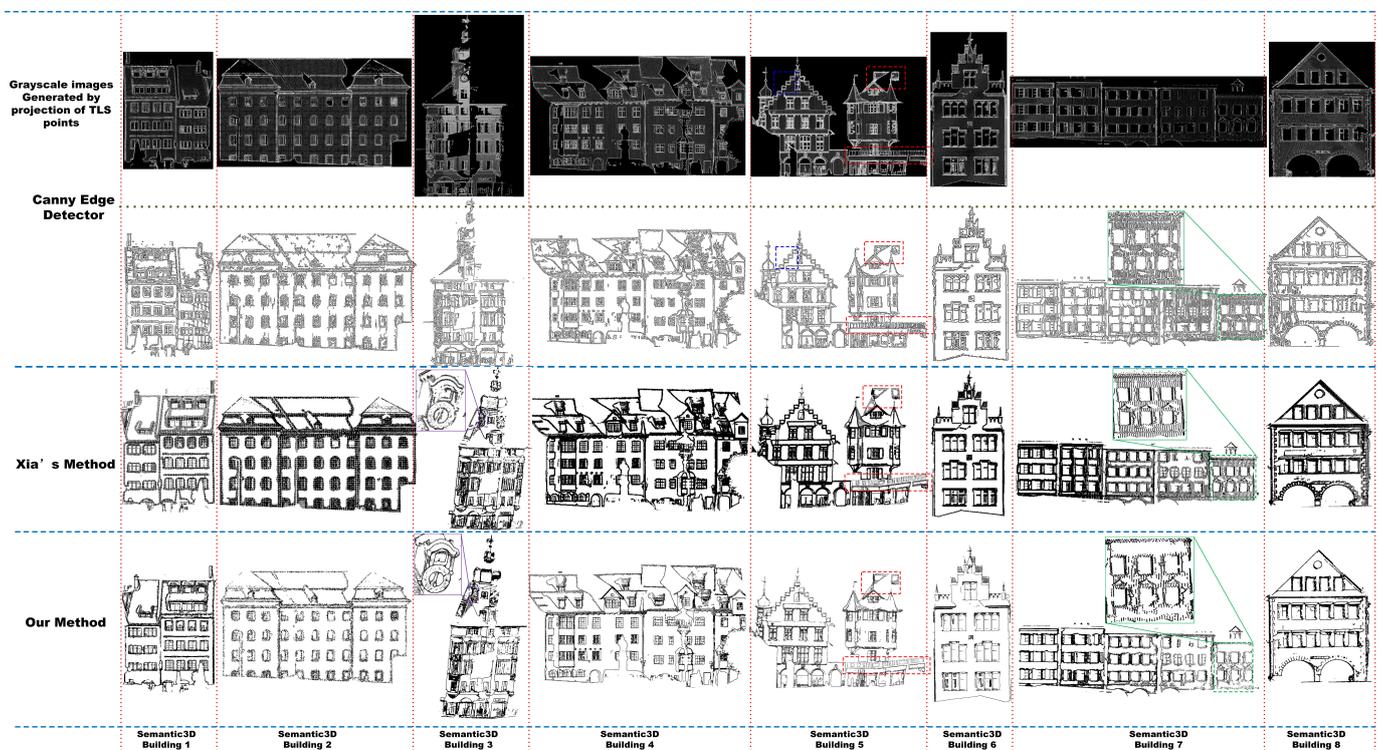


Figure 10. Comparison with classic Canny edge detector [33] and Xia's method [36] on eight representative buildings in the Semantic3D dataset. The first two rows represent results derived from the Canny edge detector. The grayscale images in the first row generated by projection of Semantic3D TLS point clouds are provided as inputs to the Canny edge detector. Xia's results are presented in the middle row, and our results are given in the bottom row.

Apart from qualitative comparisons with the Canny detector and Xia's method, we also conduct a quantitative comparison with Xia's method using eight buildings in the Semantic3D dataset. We first calculate each building's *Mean*, *SD*, *RMSE*, and *Compactness* according to the reference. Next, we average all of them to obtain the overall values of the entire Semantic3D dataset. The statistics are listed in Table 4, from which we can see that our method outperforms Xia's method in terms of geometric accuracy and compactness.

Table 4. Quantitative comparison between our method and Xia's method [36] of eight buildings in the Semantic3D dataset in terms of *Mean*, *SD*, *RMSE*, and *Compactness*. Note that all values are the average of eight buildings in the Semantic3D dataset.

Dataset	Mean (m)		SD (m)		RMSE (m)		Compactness (%)	
	Our Method	Xia's Method	Our Method	Xia's Method	Our Method	Xia's Method	Our Method	Xia's Method
Semantic3D TLS	0.0574	0.0626	0.1507	0.1771	0.1618	0.1889	18.33%	56.09%

3.6. Robustness

Apart from using the Semantic3D TLS dataset, we also employ various types of ubiquitous point clouds captured by different platforms with different LiDAR sensors to verify the robustness of the proposed algorithm. We use two patches of ALS building clouds clipped from DublinCity and Dutch AHN3 datasets. Due to the advantages of airborne scanning, the acquired building point clouds are relatively dense for rooftops and sparse and non-uniform distributed for building façades. The proposed algorithm successfully extracts the rooftop skeleton (see the enlarged rectangle views in Figure 11); however, it is hard to extract feature contours from building façades due to very sparse and irregular points and/or large-scale missing data caused by occlusion and self-occlusion.

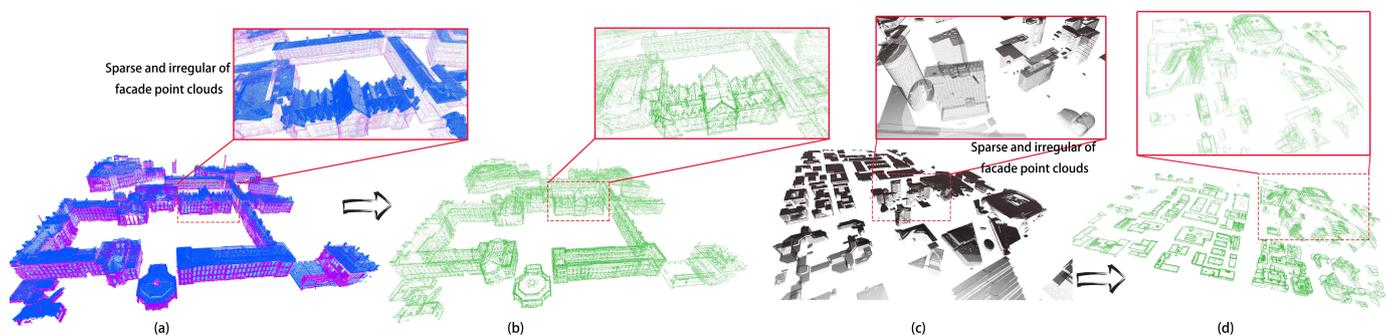


Figure 11. Critical point extraction from part of DublinCity and Dutch AHN3 datasets. Subfigures (a,b) are the part of DublinCity building point clouds and contour point results, while (c,d) are some selected building point clouds from the Dutch AHN3 dataset and their contour points. Since these two datasets are captured by airborne laser scanners, they have more detailed rooftop shapes than façades.

We used large-scale MLS point clouds from the TerraMobilita/iQmulus 3D urban analysis benchmark to test whether the proposed algorithm is sensitive to the irregularity of point clouds. Generally, for MLS point clouds, the point density at the bottom of street façades is higher than the point density at the areas of the top façades. In addition, when scanning the street façades, the laser beam can be easily blocked by moving cars, pedestrians, and street trees in front of the façades, thereby causing missing data. Despite this, the building façade contour is successfully extracted, as shown in Figure 12.

The UAV-based photogrammetric point cloud is used to test the algorithm's applicability for processing data captured from the consumer-grade UAV DJI Phantom 4 RTK. The density of UAV-based point clouds is restricted to the resolution of the acquired images. The obtained point clouds are calculated by stereo-image matching rather than directly surveying using the LiDAR technique. Because of this, the photogrammetric point clouds

miss fine details and sharp edges with massive artifacts, as demonstrated in the enlarged views in Figure 6. Despite this, the proposed algorithm has the capability to abstract the building façade contours.

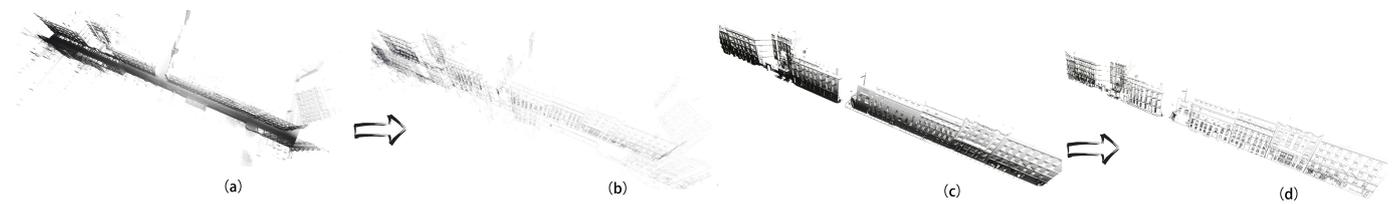


Figure 12. Critical point results of TerraMobilita/iQmulus 3D urban analysis benchmark. Subfigures (a,c) are the raw point clouds, while (b,d) are the corresponding building contour results. Point clouds in (c) come from subfigure (a). Note that subfigure (b) only includes building critical points, and the ground, trees, and other irrelevant objects are excluded according to the class label.

In addition, GeoSLAM ZEB-HORIZON point clouds are used for testing the sensitive-ness to point precision. In Figure 7, we observe that GeoSLAM point clouds have very low point precision with mess point cloud distribution. In this case, the sharp façade features cannot be clearly described in the raw point clouds; however, fortunately, the proposed algorithm can perceive the differences of the façade points and successively detect the feature point clouds, as evident in the overlap view in Figure 7c.

4. Conclusions and Suggestions for Future Works

In this paper, we present a method for extracting building façade contours. Our method analyzes each point gradient and its derived gradient structure tensor to obtain the gradient distribution for each point in the local neighborhood areas. To relieve the dependence on thresholds of gradient and gradient structure tensor, we refine the initial critical point set, striking a balance between geometric accuracy and compact representation/abstraction. We use multiple ubiquitous LiDAR datasets to verify the applicability of the proposed algorithm.

Although promising results are achieved, it should be noted that our algorithm is effective when it is applied to building contouring. The contouring results may not reasonably be expected to other objects, such as trees and cars. This requires us to recognize individual building at the instance level from LiDAR scans through machine learning [44] and/or deep learning [45,46] methods. In addition, we should be aware that the critical points are extracted based on the analysis of the gradient structure tensor in the local areas, but these critical points jointly describe the global feature of architectural shapes. This implies that in future work, we can carry out successive research on the analysis of building structures and geometric shapes. Although we provide the critical points with fine details for the description of the building façades and rooftops, these critical points cannot maintain the topological relationships, making it difficult for further reconstruction with a boundary representation. In future work, we plan to research how to segment these critical point clouds into multiple clusters with meaningful semantics using the graph convolutional network and how to organize the clusters into watertight building models using a topology optimization technique.

Author Contributions: J.L. and S.D. analyzed the data and wrote the C++ source code. J.P. and D.C. helped in the project and study design, paper writing, and data analysis. G.X., L.W. and X.L. helped in proofreading, data acquisition, and experiment comparison. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 41971415, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20201387, in part by the Open Fund of State Key Laboratory of Remote Sensing Science under

Grant OFSLRSS202010, and it was performed while the corresponding author, Dong Chen, acted as an awardee of the 2021 Qinglan Project, sponsored by Jiangsu Province, China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets including Semantic3D, DublinCity, Dutch AHN3, and TerraMobilita/iQmulus 3D are publicly available from the corresponding benchmark websites or distribution platform. Other datasets analyzed in this study are acquired by ourselves.

Acknowledgments: The authors would like to thank Shaobo Xia for providing C++ code in [36] for comparison.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rosenfeld, A.; Thurston, M. Edge and curve detection for visual scene analysis. *IEEE Trans. Comput.* **1971**, *100*, 562–569. [[CrossRef](#)]
2. Zhang, W.; Zhang, W.; Gu, J. Edge-semantic learning strategy for layout estimation in indoor environment. *IEEE Trans. Cybern.* **2019**, *50*, 2730–2739. [[CrossRef](#)]
3. Lepora, N.F.; Church, A.; De Kerckhove, C.; Hadsell, R.; Lloyd, J. From pixels to percepts: Highly robust edge perception and contour following using deep learning and an optical biomimetic tactile sensor. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2101–2107. [[CrossRef](#)]
4. Baterina, A.V.; Oppus, C. Image edge detection using ant colony optimization. *Wseas Trans. Signal Process.* **2010**, *6*, 58–67.
5. Sadiq, B.O.; Sani, S.; Garba, S. Edge detection: A collection of pixel based approach for colored images. *Int. J. Comput. Appl.* **2015**, *113*, 29–32.
6. Marmanis, D.; Schindler, K.; Wegner, J.D.; Galliani, S.; Datcu, M.; Stilla, U. Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS J. Photogramm. Remote Sens.* **2018**, *135*, 158–172. [[CrossRef](#)]
7. Duan, R.L.; Li, Q.x.; Li, Y.h. Summary of image edge detection. *Opt. Tech.* **2005**, *3*, 415–419.
8. Guo, K.Y.; Hoare, E.G.; Jasteh, D.; Sheng, X.Q.; Gashinova, M. Road edge recognition using the stripe Hough transform from millimeter-wave radar images. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 825–833. [[CrossRef](#)]
9. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
10. Ni, H.; Lin, X.; Zhang, J.; Chen, D.; Peethambaran, J. Joint clusters and iterative graph cuts for ALS point cloud filtering. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 990–1004. [[CrossRef](#)]
11. Hackel, T.; Wegner, J.D.; Schindler, K. Contour detection in unstructured 3d point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, U.S., 26 June–1 July 2016; pp. 1610–1618.
12. Wang, R.; Peethambaran, J.; Chen, D. LiDAR point clouds to 3-D Urban Models: A review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 606–627. [[CrossRef](#)]
13. Mineo, C.; Pierce, S.G.; Summan, R. Novel algorithms for 3D surface point cloud boundary detection and edge reconstruction. *J. Comput. Des. Eng.* **2019**, *6*, 81–91. [[CrossRef](#)]
14. Ahmed, A.; Paul, M.; Pickering, M. Dynamic point cloud texture video compression using the edge position difference oriented motion model. In Proceedings of the 2021 Data Compression Conference (DCC), Snowbird, UT, USA, 23–26 March 2021; p. 335.
15. Xia, S.; Chen, D.; Wang, R.; Li, J.; Zhang, X. Geometric primitives in LiDAR point clouds: A review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 685–707. [[CrossRef](#)]
16. Li, J.; Zhao, P.; Hu, Q.; Ai, M. Robust point cloud registration based on topological graph and Cauchy weighted lq-norm. *ISPRS J. Photogramm. Remote Sens.* **2020**, *160*, 244–259. [[CrossRef](#)]
17. Gojcic, Z.; Zhou, C.; Wegner, J.D.; Guibas, L.J.; Birdal, T. Learning multiview 3D point cloud registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–20 June 2020; pp. 1759–1769.
18. Choi, C.; Trevor, A.J.; Christensen, H.I. RGB-D edge detection and edge-based registration. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 1568–1575.
19. Shi, W.; Rajkumar, R. Point-gnn: Graph neural network for 3d object detection in a point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–20 June 2020; pp. 1711–1719.
20. Hu, F.; Yang, D.; Li, Y. Combined edge-and stixel-based object detection in 3D point cloud. *Sensors* **2019**, *19*, 4423. [[CrossRef](#)]
21. Wang, R.; Ferrie, F.P.; Macfarlane, J. A method for detecting windows from mobile LiDAR data. *Photogramm. Eng. Remote Sens.* **2012**, *78*, 1129–1140. [[CrossRef](#)]
22. Song, H.; Feng, H.Y. A progressive point cloud simplification algorithm with preserved sharp edge data. *Int. J. Adv. Manuf. Technol.* **2009**, *45*, 583–592. [[CrossRef](#)]
23. Li, H.; Zhong, C.; Hu, X.; Xiao, L.; Huang, X. New methodologies for precise building boundary extraction from LiDAR data and high resolution image. *Sens. Rev.* **2013**, *33*, 157–165. [[CrossRef](#)]

24. Poullis, C. A framework for automatic modeling from point cloud data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2563–2575. [[CrossRef](#)] [[PubMed](#)]
25. Wang, Y.; Ewert, D.; Schilberg, D.; Jeschke, S. Edge extraction by merging 3D point cloud and 2D image data. In Proceedings of the 2013 10th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT), Melville, NY, USA, 21–22 October 2013; pp. 1–6.
26. Li, Y.; Wu, H.; An, R.; Xu, H.; He, Q.; Xu, J. An improved building boundary extraction algorithm based on fusion of optical imagery and LiDAR data. *Optik* **2013**, *124*, 5357–5362. [[CrossRef](#)]
27. Chen, Y.; Zhang, W.; Zhou, G.; Yan, G. A novel building boundary reconstruction method based on lidar data and images. In *International Symposium on Photoelectronic Detection and Imaging 2013: Laser Sensing and Imaging and Applications*; SPIE-International Society for Optics and Photonics: Bellingham, WA, USA, 2013; Volume 8905, p. 890522.
28. Yang, B.; Zang, Y. Automated registration of dense terrestrial laser-scanning point clouds using curves. *ISPRS J. Photogramm. Remote Sens.* **2014**, *95*, 109–121. [[CrossRef](#)]
29. Demarsin, K.; Vanderstraeten, D.; Volodine, T.; Roose, D. Detection of closed sharp edges in point clouds using normal estimation and graph theory. *Comput. Aided Des.* **2007**, *39*, 276–283. [[CrossRef](#)]
30. Borges, P.; Zlot, R.; Bosse, M.; Nuske, S.; Tews, A. Vision-based localization using an edge map extracted from 3D laser range data. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 4902–4909.
31. Ni, H.; Lin, X.; Ning, X.; Zhang, J. Edge detection and feature line tracing in 3d-point clouds by analyzing geometric properties of neighborhoods. *Remote Sens.* **2016**, *8*, 710. [[CrossRef](#)]
32. Wang, R.; Lai, X.; Hou, W. Study on edge detection of LIDAR point cloud. In Proceedings of the 2011 International Conference on Intelligent Computation and Bio-Medical Instrumentation, Wuhan, China, 14–17 December 2011; pp. 71–73.
33. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, 679–698. [[CrossRef](#)]
34. Pauly, M.; Mitra, N.J.; Giesen, J.; Gross, M.H.; Guibas, L.J. Example-Based 3d Scan Completion. 2005. Available online: https://infoscience.epfl.ch/record/149337/files/pauly_2005_EBS.pdf (accessed on 6 August 2021).
35. Harris, C.G.; Stephens, M. A Combined Corner and Edge Detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September, 1988; pp. 1–6. Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.434.4816&rep=rep1&type=pdf> (accessed on 6 August 2021).
36. Xia, S.; Wang, R. A fast edge extraction method for mobile LiDAR point clouds. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1288–1292. [[CrossRef](#)]
37. Pauly, M.; Gross, M.; Kobbelt, L.P. Efficient simplification of point-sampled surfaces. In Proceedings of the IEEE Visualization, VIS 2002, Boston, MA, USA, 27 October–1 November 2002; pp. 163–170.
38. Huang, H.; Li, D.; Zhang, H.; Ascher, U.; Cohen-Or, D. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph.* **2009**, *28*, 1–7. [[CrossRef](#)]
39. Lipman, Y.; Cohen-Or, D.; Levin, D.; Tal-Ezer, H. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph.* **2007**, *26*, 22–es. [[CrossRef](#)]
40. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. *arXiv* **2017**, arXiv:1704.03847.
41. Du, J.; Chen, D.; Wang, R.; Peethambaran, J.; Mathiopoulos, P.T.; Xie, L.; Yun, T. A novel framework for 2.5-D building contouring from large-scale residential scenes. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4121–4145. [[CrossRef](#)]
42. Zolanvari, S.; Ruano, S.; Rana, A.; Cummins, A.; da Silva, R.E.; Rahbar, M.; Smolic, A. DublinCity: Annotated LiDAR point cloud and its applications. *arXiv* **2019**, arXiv:1909.03613.
43. Vallet, B.; Brédif, M.; Serna, A.; Marcotegui, B.; Papanoditis, N. TerraMobilita/iQmulus urban point cloud analysis benchmark. *Comput. Graph.* **2015**, *49*, 126–133. [[CrossRef](#)]
44. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
45. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
46. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–20 June 2020; pp. 11108–11117.