Article

Morphological Operations to Extract Urban Curbs in 3D MLS Point Clouds

Borja Rodríguez-Cuenca [1,†]**, Silverio García-Cortés** [2,†]**, Celestino Ordóñez** [2,*,†] **and María C. Alonso** [1,†]

[1] Department of Physics and Mathematics, Alcalá University, Campus Universitario Ctra. Madrid-Barcelona, km. 33,600, Alcalá de Henares 28871, Spain; borja.rodriguezc@edu.uah.es (B.R.-C.); mconcepcion.alonso@uah.es (M.C.A.)

[2] Department of Mining Exploitation and Prospecting, University of Oviedo, Escuela Politécnica de Mieres, C/Gonzalo Gutiérrez Quirós, Mieres 33600, Spain; sgcortes@uniovi.es

* Correspondence: ordonezcelestino@uniovi.es; Tel.: +34-985-458-027

† These authors contributed equally to this work.

Abstract: Automatic curb detection is an important issue in road maintenance, three-dimensional (3D) urban modeling, and autonomous navigation fields. This paper is focused on the segmentation of curbs and street boundaries using a 3D point cloud captured by a mobile laser scanner (MLS) system. Our method provides a solution based on the projection of the measured point cloud on the XY plane. Over that plane, a segmentation algorithm is carried out based on morphological operations to determine the location of street boundaries. In addition, a solution to extract curb edges based on the roughness of the point cloud is proposed. The proposed method is valid in both straight and curved road sections and applicable both to laser scanner and stereo vision 3D data due to the independence of its scanning geometry. The proposed method has been successfully tested with two datasets measured by different sensors. The first dataset corresponds to a point cloud measured by a TOPCON sensor in the Spanish town of Cudillero. The second dataset corresponds to a point cloud measured by a RIEGL sensor in the Austrian town of Horn. The extraction method provides completeness and correctness rates above 90% and quality values higher than 85% in both studied datasets.

Keywords: curbs; road boundaries; 3D point cloud; segmentation; feature extraction

1. Introduction

Laser scanner sensors and stereo vision systems provide fast and accurate three-dimensional (3D) information on objects, buildings, and landscapes without maintaining direct contact with the measured objects. This information is useful in several remote sensing applications like digital terrain model generation [1], 3D city modeling [2], and feature extraction [3]. Laser scanner sensors can be placed on aerial (aerial laser scanners, ALS) or terrestrial platforms (terrestrial laser scanners, TLS). TLS can be categorized into two types: static and dynamic. Static TLS data collection is carried out from base stations: A sensor is fixed in a base station, from which the point cloud is acquired. Dynamic TLS or mobile laser scanner (MLS) sensors are installed in a mobile platform. MLS sensors have a navigation system based on global navigation satellite systems (GNSS) and inertial measurement units (IMU). This work is focused on segmenting 3D point clouds produced by MLS sensors to determine existing curbs in urban environments. An accurate method for determining the location of curbs, road boundaries, and urban furniture is crucial for several applications, including 3D urban modeling and developing autonomous navigation systems [4,5]. Moreover, accurate and automatic

detection of cartographic entities saves a great deal of time and money when creating and updating cartographic databases [6]. The current trend in remote sensing feature extraction is the development of methods as automatic as possible. The aim is to develop algorithms that can obtain accurate results with the least possible human intervention in the process. It is difficult to create a fully automatic method of determining the location of and extracting every piece of urban furniture in a city for the following reason: Urban furniture has a heterogeneous typology; every city, and almost every street, has its own typical furniture. Most works on feature extraction have proposed semi-automatic methods in which the user must control a few settings for accurate detection. The authors of this work have developed a semi-automatic method to detect street curbs through segmenting the measured point cloud. Furthermore, a method to extract upper and lower curb edges is proposed. We attempted to minimize the number of thresholds. In the current method, the user must control three settings that depend directly on features of the studied area: the height of the curb, the point density in the curb's vertical wall, and the roughness. The paper is organized as follows: Section 2 summarizes previous studies related to ours; Section 3 shows the proposed method to segment the point cloud; and in Section 4 the results obtained in two study cases performed with different MLS sensors are detailed. Finally, our conclusions and future lines of work are described in Section 5.

2. Related Work

Many applications for TLS have been reported since the appearance of these systems. The 3D modeling of buildings and indoor areas [7,8], the geometry verification of tunnels [9], the detection of urban furniture and pole-like objects [10], and the modeling and reconstruction of 3D trees [11] are some of the applications for which TLS sensors have been used. Additionally, several applications for point clouds detected via MLS sensors exist in the current literature. They have been used in applications such as vertical wall extraction [12], façade modeling [13], building footprint detection [14], and the extraction of pole-like objects, such as traffic signs, lamp posts, or tree trunks [15,16].

The amount of point clouds data provided by laser scanning systems is extremely large, composed of (x, y, z) coordinates and additional information such as intensity, Global Positioning System (GPS) time, or the scanning angle of millions of points. The analysis and processing of these data is computationally complex. Hence, in order to reduce the processing times and the complexity of the datasets, the point clouds are often simplified before an algorithm is used for feature extraction, mapping, or decision-making. In some cases, the point clouds are segmented into several clusters that are individually analyzed and classified [17]. In [18] a segmentation method based on the difference of normals of a point and its neighborhood applied in a multi-scale approach is proposed. The difference of normals algorithm is efficient for segmenting a large 3D point cloud into various objects of interest at different scales such as cars, road curbs, trees, or buildings. In other works, the 3D point cloud is divided into other smaller clouds formed by slices of the original to reduce the amount of data. In [19], as a previous step in the detection of existing street curbs, the measured point cloud is divided into several road cross sections using the GPS time data. Another option to make the point cloud more manageable is to decompose the measured data in a 3D voxel grid. In [20], a method is presented that performs a 3D scene analysis from streaming data. This procedure consists of a hierarchical segmentation formed by multiple consecutive segmentations of the point cloud. The original point cloud is sparsely quantized into infinitely tall pillars; each pillar is quantized into coarse blocks and each block contains a linked list of its occupied voxels. In this work, voxels are considered the atomic unit that object categories are assigned to. The original point cloud is projected into a 2D raster image that represents the XY surface. Thus, the 3D information is reduced to a 2D raster in which image processing techniques can be applied to determine the location of the target features.

The work presented in this paper is devoted to curb detection from MLS point clouds. In the current literature, there are many studies related to this issue. Some of them use as input data point clouds obtained from stereo vision; recently, several authors have focused on the detection of road markings, lines, and roadsides in straight and curved areas based on data obtained by stereo

cameras [21,22]. There are also some methods in the current literature used to detect curbs and roadsides based on point clouds measured with TLS and MLS sensors. In [23], a method to detect curbs using 3D scanner sensor data was presented. The detection process starts with the voxelization of the point cloud and the separation of those points that represent the ground. Later, candidate points for curbs are selected based on three spatial variables: height difference, gradient value, and normal orientation. This gradient value, that is, the local elevation rate of change, is obtained by applying a 3×3 Sobel operator in both horizontal and vertical directions in a 2D elevation map. Using a short-term memory technique, every point located in a voxel whose vertical projection is in the road is considered a false positive and is deleted. Finally, the curb is detected by adjusting a parabolic model to the candidate points and performing a RANSAC algorithm to remove false positives. The performance of the method depends on the correct selection of the thresholds for each of the three variables used. This method provided a detection rate of about 98% in two studied datasets.

Weiss and Dietmayer [24] automatically determined the position of lane markings, sidewalks, reflection posts, and guardrails by a vertically and horizontally automotive laser scanner data. Curb detection applies a third-order Gaussian filter to sharpen the vertical distance profile, which defines the shape of the curb. This profile is divided into sections with a certain width, forming an accumulative histogram. Candidate curbs are found through a histogram-based algorithm that searches those slots of the histogram that are candidates to represent curbs and guardrails. Because not every candidate is a valid curb, the locations of real curbs are determined by analyzing the heights, slopes, and interruptions of every polygon.

Belton and Bae [25] proposed a method to automatize the identification of curbs and signals using a few steps. The rasterization of the 3D point cloud into a 2D grid structure allows each cell to be examined separately. First, the road is extracted; then, cells that are adjacent to the road are likely to contain curbs. Points in these cells are used to determine the vertical plane of the curb, from which a 2D transversal section is calculated. The top and bottom of the curb are determined as the points that are furthest above and below the line defined by the two furthermost points in the 2D section. This procedure has several limitations. The proposed method would not provide good results detecting concave and non-horizontal roads; furthermore, the method could provide poor results for shorter or curved curbs due to confusing edges with other points of the studied profile.

Yang *et al.* [19] carried out edge detection by dividing the measured point cloud into 2D sections using the GPS time at which every point was registered. They applied a moving window to these 2D sections to detect the roads and road boundaries. Curbs were detected by analyzing the elevation and shape change in the moving windows studied. They also presented a method to detect curbs in occluded parts of the cloud, but some problems in areas with irregular shapes were detected. The value of the parameters and the length of the moving window are critical to the performance of the proposed method.

A recent work by Hervieu and Soheilian [26] describes a method to extract curbs and ramps, as well as reconstruct lost data in areas hidden by obstacles in the street. A system for the reconstruction of road and sidewalk surfaces is also proposed. They adjust a plane to a group of points from the cloud and compute the angular distance between the normal vector and the z vector. After that, a prediction/estimation model is applied to detect road edges. The procedure requires the user to manually select the curb direction, which is not always easy. This method could fail in curved or occluded sections. To solve this problem, they propose a semi-automatic solution in which the user must choose some points of the non-detected curb to reconstruct these sections.

In [27], Kumar *et al.* developed a method to detect road boundaries in both urban and rural roads, where the non-road surface is comprised of grass and soil and the edges are not as easily defined by slope changes alone. A 2D rasterization of the slope, reflectance, and pulse width of the detected point cloud is carried out. Gradient vector flow and a balloon model are combined to create a parametric active contour model, which allows the road boundaries to be determined. Roadside detection is carried out using a snake curve, which is initialized based on the navigation track of a mobile van

along the road section. The snake curve moves using an iterative process until it converges on the roadsides, where the minimum energy state is located. This method has been tested in straight sections and provided good results, but its performance in curved sections is unknown. The procedure is computationally complex, which could make the detection process too slow.

Serna and Marcotegui [28] propose a method to create obstacle maps from MLS point clouds. They construct range images projecting 3D points and use morphological filters to remove isolate and not elongated structures. Our approach is in some way similar to that, although there are some significant differences since we do not use the minimal range image to detect the lowest points, and morphological operators are applied in a different way. We also add density and roughness as parameters to detect the curbs.

Apart from the methods described in the current literature, there are other solutions for curb detection in commercial software packages [29], but their technical details could not be found in the literature. These solutions are not fully automatic; users must provide some initial information to the software.

3. Methodology

The method proposed in this work uses 3D point coordinates (*x*, *y*, *z*) of the cloud data measured by MLS as inputs. The output of the procedure is a new 3D point cloud formed by those points that belong to a curb. The proposed method for segmenting MLS data consists of four consecutive steps, as shown in Figure 1.
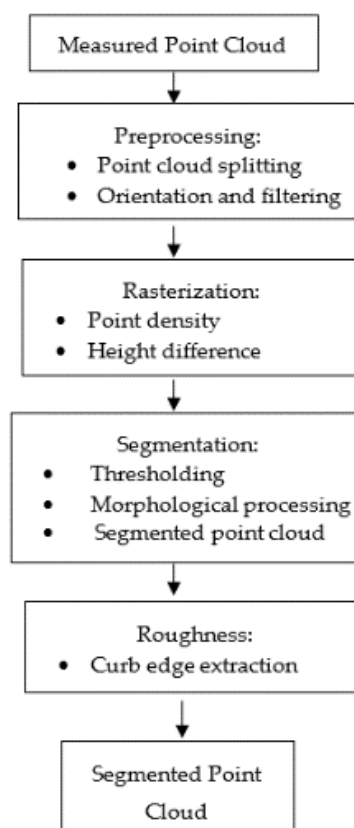

Figure 1. Flow chart of the proposed method.

3.1. Preprocessing

The 3D point clouds data are difficult to process due to the large amount of information corresponding to millions of points. The work with these raw MLS data could involve a significant

amount of time and computational effort. Thus, in this first stage, some methods to facilitate the handling of the point cloud are carried out. Preprocessing is divided into three main steps: a point cloud splitting, a coordinate system change, and a filtering phase.

3.1.1. Point Cloud Splitting

We divide the measured point clouds into slices and process them individually. In the case of Test Site 1, the original point cloud is divided into four slices, each formed by 1.5 million points (Figure 2). The following steps of the proposed method are applied on every slice separately.
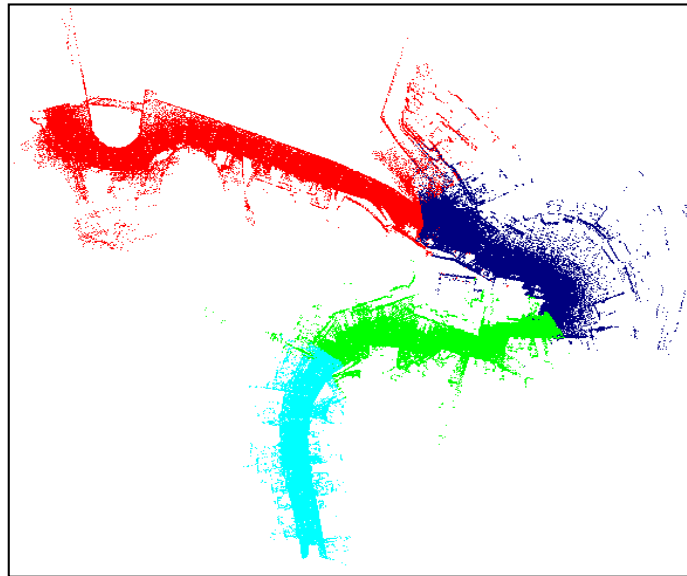

Figure 2. Original MLS point cloud in dataset 1 divided into four sections (represented by different colors).

3.1.2. Orientation and Filtering

In this step, a coordinate system change is carried out, moving from a global coordinate system, in which the MLS measures the point cloud, to a local coordinate system. The point clouds registered by the MLS sensor are properly geo-referenced in a global reference system by means of a navigation system and an inertial measurement unit (IMU), which provides coordinates within a global frame to every point. The point clouds used in this work are geo-referenced using UTM projection in Zone 29 for dataset 1 and UTM projection in Zone 33 for dataset 2, and WGS84 ellipsoidal heights in both cases. MLS datasets are usually quite large, and managing them is computationally difficult and expensive work. In order to ease and speed up every operation in the segmentation procedure, the original coordinate system is transformed by means of a translation and three rotations into a local Cartesian coordinate system located at the beginning of the MLS trajectory and in which the x-axis is coincident with the average direction of the vehicle.

This procedure is carried out individually for every section in which the original point cloud was divided in the former step. Once the coordinate system is changed, a filtering process is then carried out to reduce the point cloud size. Every point located under the GPS antenna has a negative height, and it is assumed that those points that represent curbs are below the antenna. For this reason, only the points with local negative z-coordinate values are kept, and the points located over the GPS antenna height are removed. Figure 3 shows every step of the preprocessing stage on slice 1 of Test Site 1.
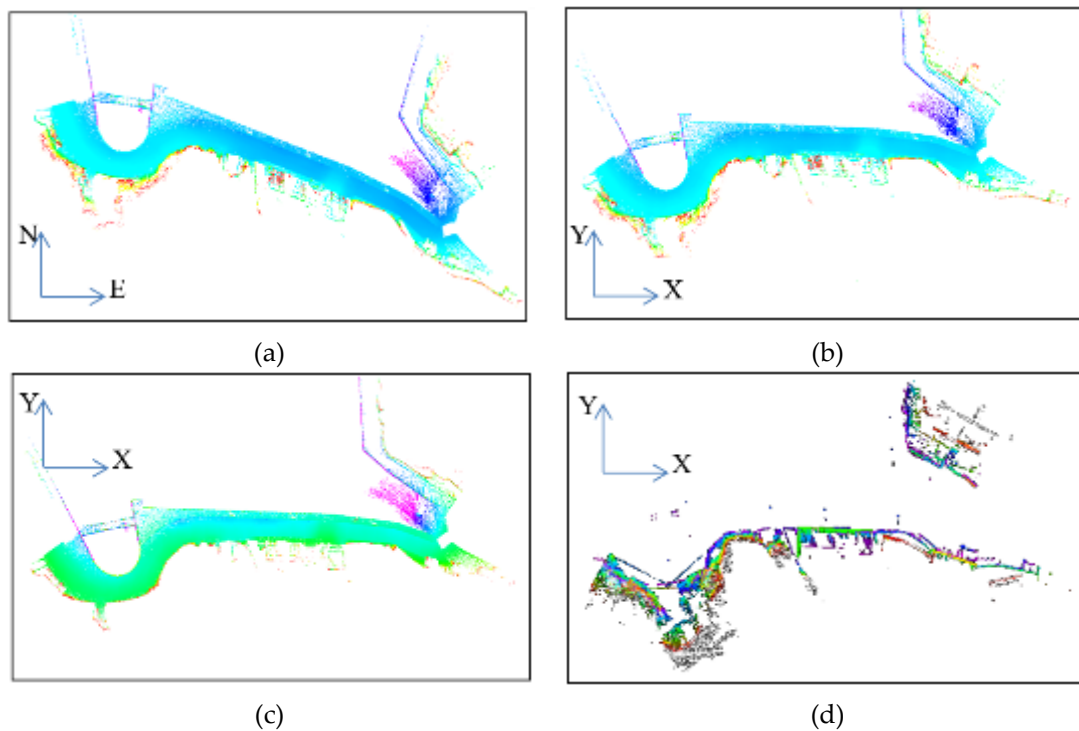
Figure 3. (**a**) Point cloud corresponding to slice 1 in Test Site 1 in UTM coordinate system; (**b**) after the coordinate system change; (**c**) filtering in negative; and (**d**) positive heights.

3.2. Rasterization

In this step, to simplify the analysis and reduce the computing cost of segmentation, the 3D point cloud is projected in a 2D grid that represents the XY surface, moving from a 3D cloud to a 2D raster image. Moreover, in a raster image, it is possible to apply image analysis techniques to detect those pixels that contain points that belong to a roadside. The efficiency of the detection process will depend on the pixel size of the created image. The cell size also depends on the point density of the point cloud: the higher the point cloud's density, the higher the image resolution can be, but this can also require more computation resources. In any case, the grid spacing must be large enough to contain a significant number of points, but small enough to allow only a small number of salient features in each cell [25]. An empirical rule has been created to determine the optimum cell size of the grid for the rasterization of every point cloud. This rule depends on the distance between consecutive scans in the central part of the detected point cloud. Thus, the cell size must be between 4 and 5 times larger than the distance between consecutive scans. We have found that these values allow the proper detection of curbs with different scan densities and curb widths. For each pixel, two values are calculated and saved as digital values (DVs) in the grid: the difference between the highest and the lowest point of all points contained in the studied cell (resulting in an image similar to a normalized digital surface model (nDSM), and the number of points contained in every cell. Thus, the rasterization step provides two images: one with DVs representing the height difference between the highest and the lowest points (referred to as a nDSM from now on) and another in which each pixel's DV is the number of points contained within it (referred to as image density from now on). Normally, the intensity of the laser reflected pulse is also available in MLS data, but we do not consider that information relevant for this work.

3.3. Segmentation

This step consists of segmenting the rasterized 3D point cloud in order to select those pixels of the image that are candidates to represent a curb. It is divided into two stages: thresholding and

morphological operations. These two stages provide a binary 2D image; the points of the measured cloud that are contained within the one-value pixels are extracted and considered curbs.

3.3.1. Thresholding

There are several features in a point cloud data that can be used to segment the cloud or for feature extraction issues, such as the normal and the curvature or the reflected laser intensity, among others. For curb segmentation, two features of the real curb feature were taken into account: the difference between the highest and the lowest point of a curb and the point density in these elements. Curbs have a certain height (normally higher than 5 cm and lower than 25 cm) and a higher point density than horizontal surfaces due to their vertical face. Figure 4 shows the value of the considered features for four different urban elements in the negative high point cloud: curbs, roads, façades, and sidewalks. Surfaces that are orthogonal to the laser pulses (façades and curbs) show a higher density than those that are nearly parallel to the laser pulses (roads and sidewalks) [30]. The point density in curbs is higher than in horizontal elements like roads and sidewalks, but lower than higher surfaces like façades. Furthermore, the height difference in curbs is much lower than other vertical elements as the façades, but a bit higher than horizontal classes.
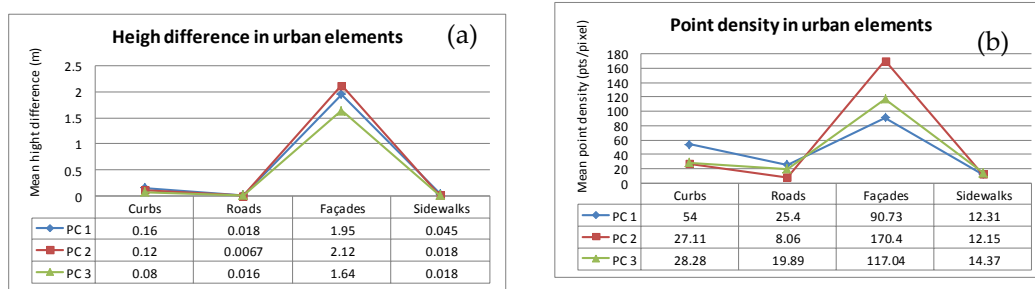

Heigh difference in urban elements (a)

	Curbs	Roads	Façades	Sidewalks
PC 1	0.16	0.018	1.95	0.045
PC 2	0.12	0.0067	2.12	0.018
PC 3	0.08	0.016	1.64	0.018

Point density in urban elements (b)

	Curbs	Roads	Façades	Sidewalks
PC 1	54	25.4	90.73	12.31
PC 2	27.11	8.06	170.4	12.15
PC 3	28.28	19.89	117.04	14.37

Figure 4. Height difference (**a**) and mean point density (**b**) of four urban elements in three different point clouds (PC).

Those pixels in the nDSM image whose digital values fall between two established thresholds (Hmin and Hmax in Equation (1)) and whose density image DV is higher than a minimum threshold (Dmin) (Equation (2)) are selected as curb-candidate pixels. This information is saved in a binary image in which pixels are labeled with 1 if they represent curb-candidate pixels and 0 otherwise.

$$Hmin \; < \; nDSM\,[i,j] \; < \; Hmax \tag{1}$$

$$Density\;image\;[i,j] \; > \; Dmin\;(points/pixel) \tag{2}$$

Pixels corresponding with the background (valued 0) will not be taken into account in the following step of the detection method. There will be more pixels, in addition to those curbs, that fulfill the conditions imposed and should be eliminated via the following process.

3.3.2. Morphological Operations

A morphological opening operator [31] is applied over the binary image to remove those groups of pixels that do not represent a curb. In the dilation operation, every pixel that is connected with the candidate pixels of the binary image is added to the set of curb candidate pixels. This morphological operation is applied in a 3×3 window. The results of every step of the method can be seen in Figure 5.
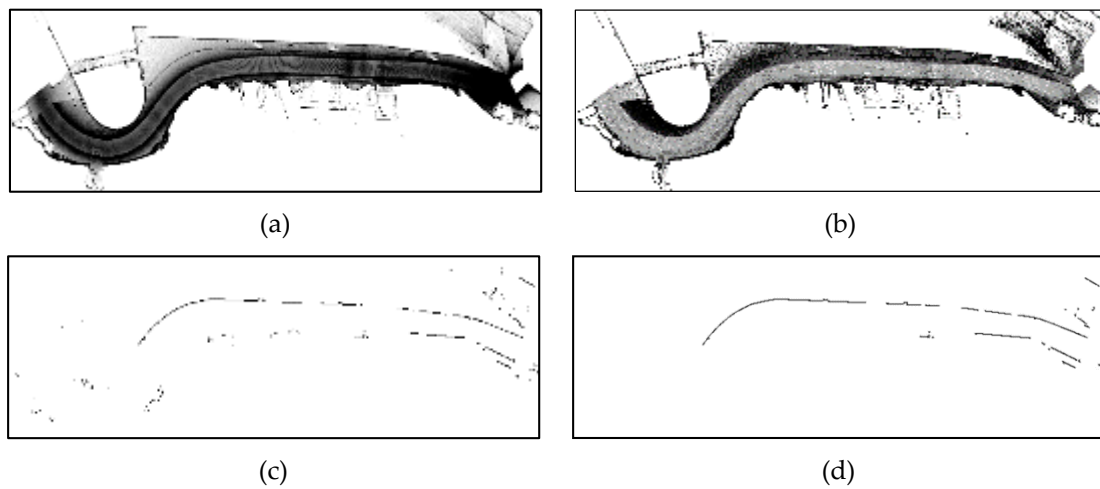
Figure 5. Results obtained in every step of the proposed method to detect curbs for slice 1 of dataset 1: (**a**,**b**) are the density and the nDSM image, and (**c**,**d**) are the results of the thresholding and the morphological operations, respectively.

3.3.3. Segmented Point Cloud

The final goal of this method is the segmentation of the existing curbs in the point cloud. Thus, once the curb detection in the binary image is carried out, it is time to move from the 2D raster image to the 3D point cloud. The new point cloud is composed of only those points belonging to the negative Z-value cloud and whose (x, y) coordinates are within the one-value pixels of the binary image obtained in the former step. The segmented point cloud is in the local coordinate system. In order to combine the measured point cloud and the segmented one, the original global coordinate system is recovered and both clouds are superimposed (Figure 6).
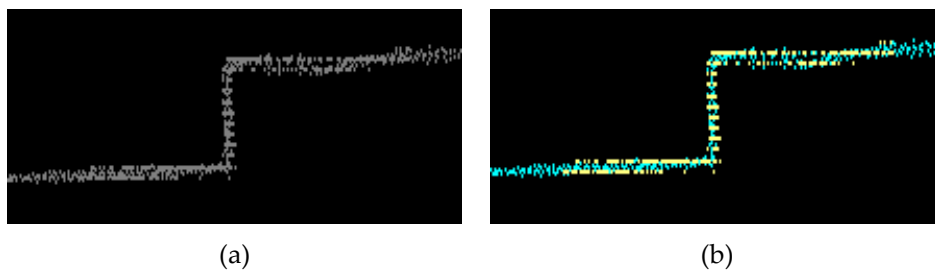

Figure 6. (**a**) Profile of a curb in the measured point cloud and (**b**) the segmented curb, with the detected curb in yellow and the background points in blue.

3.4. Roughness

The extraction of upper and lower curb edges is based on the roughness of the point cloud. In this study, the roughness (R_i) is computed for every point. This value is measured from those points contained in a sphere of radius r centered on the studied point (P_i). It is defined as the distance between the studied point P_i and the least square best fitting plane comprising P_i and its neighborhood points inside the radius r sphere [32]. This is similar to the Locally Fitted Surfaces (LoFS) implemented in [33]. The lowest roughness values correspond to flat surfaces while higher roughness values take place in those elements with irregular shapes [16]. It has been observed that, at certain neighborhoods' size, points that belong to curb edges take higher roughness values than those points that represent vertical curb walls. As can be seen in Figure 7, for a neighborhood size close to curb height (between 10 and 15 cm), roughness in curb edges is far different than in curb walls. However, in large neighborhoods

(above 25 cm) both edges and curb walls present a similar performance, making it difficult to discern between these classes in this situation. Thus, a threshold (R_{edges} in Equation (3)) has been established to extract those points that represent curb edges. The result of the upper and lower edge extraction in a section of Test Site 1 can be seen in Figure 8.
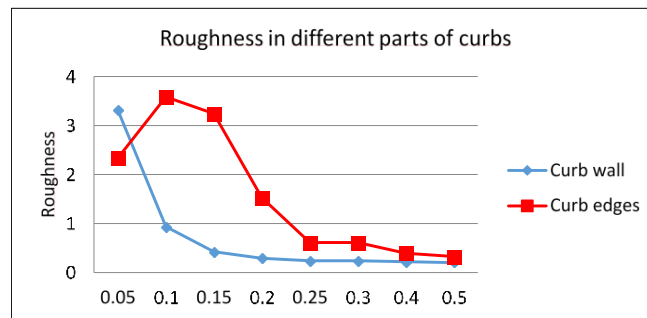
$$Roughness(R_i) > R_{edges} \tag{3}$$

Figure 7. Roughness values in curb walls and curb edges (*vertical* axis) for different neighborhood sizes (*horizontal* axis, in cm).
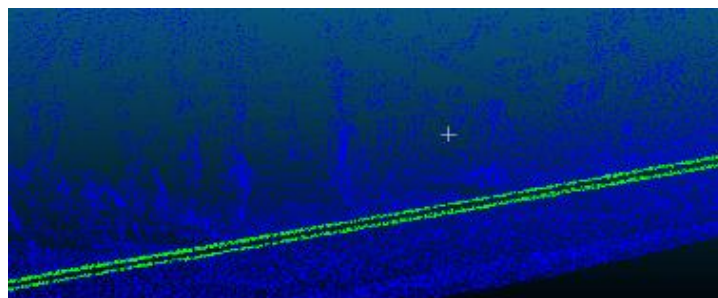

Figure 8. In green, upper and lower curb edges detected in Test Site 1.

4. Results and Discussion

4.1. Test Site 1: Curb Representation and Accuracy Evaluation

The point cloud from Test Site 1 was measured in Cudillero, a town in the north of Spain. This is a typical fishing village with low houses, very narrow streets, sometimes only one-way, and very tight sidewalks. This point cloud covers a 400-m street with both straight and curved sections (Figure 9).
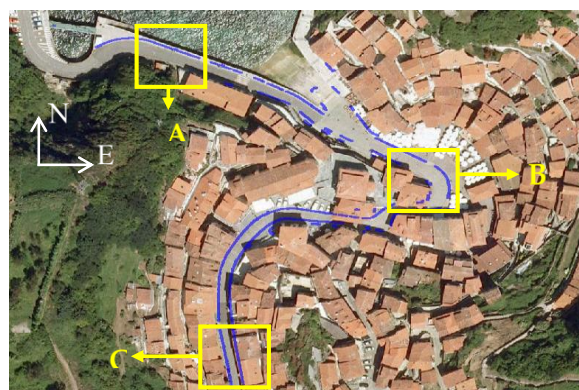

Figure 9. Aerial photograph and location of Test Site 1.

The presence of building stone walls, trucks, and other obstacles, such as fences or bar terraces, makes roadside detection more difficult. The measured point cloud comprises more than 6 million points. The values of the algorithm settings used in Test Site 1 are listed in Table 1. In the current work the pixel size was set to 20 cm × 20 cm because the mean distance between consecutive scan lines was approximately 4 cm. Regarding the height difference, the Hmin was set to 0.05 m and Hmax to 0.20 m. The minimum point density was set to 20 points per pixel and parameters related to roughness value and neighborhood size were fixed at 2.5 and 12 cm, respectively, in both test sites.

Table 1. Algorithm settings used at Test Site 1.

	Algorithm Settings
Pixel size	20 cm × 20 cm
Δh (Hmin and Hmax)	$0.05 <$ nDSM < 0.20
Minimum point density (Dmin)	20 points/pixel
Roughness (Redges)	2.5
Roughness neighborhood	12 cm

For a visual analysis of the results obtained in Test Site 1, the detected curbs have been superimposed on an ortho-image of the studied area. The results of the curb edge segmentation method are thoroughly shown in three sections: A, B, and C. Detail A represents a road section with a sidewalk on one side and a stone wall on the other side, with no sidewalk. Detail B is a curved section of a street, and Detail C is focused on a street with sidewalks on both sides of the road. For each detail, three images are shown: a street-level view (Figure 10a,d,g), the original point cloud of the detail (Figure 10b,e,h), and the result of the segmentation, in which curbs are represented in yellow and the background in blue (Figure 10c,f,i). It can be seen that the proposed method correctly detected the existing curbs both in straight and curved sections.
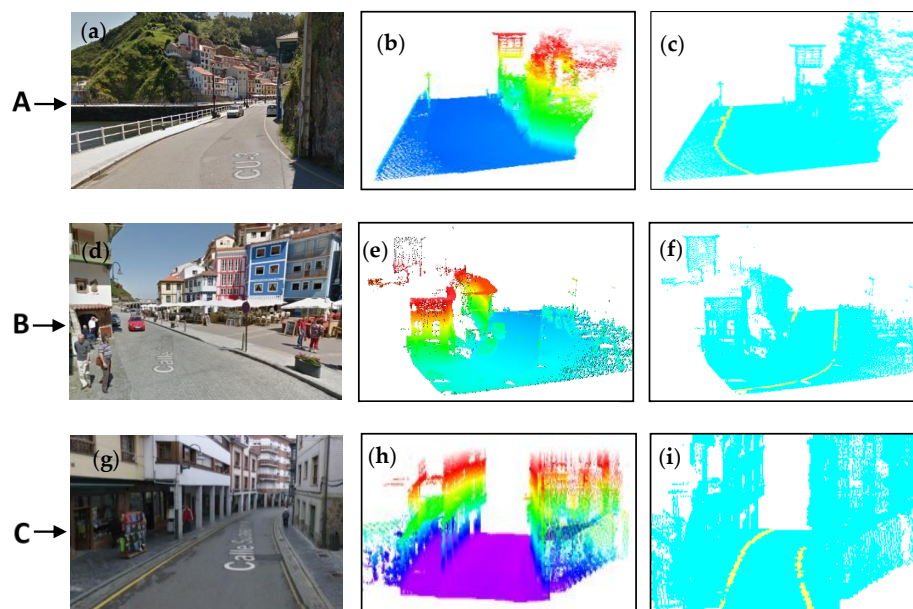

Figure 10. (**a**–**i**)Three detailed views (A, B and C) of Test Site 1. The first column shows the street-level imagery (Figure 10a,d,g). The second column represent the original registered point cloud (Figure 10b,e,h). The results of the detection developed method are shown in the third column (Figure 10c,f,i).

In order to evaluate the accuracy of the curb detection, a manual extraction of the road boundaries from the original point cloud was carried out. It was performed by digitizing the observed road borders from the point cloud as the ground truth data. For Test Site 1, the ground truth has a length of almost 520 m. The evaluation of the results was carried out by comparing the curbs extracted by

the proposed method with the previously compiled ground truth. This was performed using three indices commonly used in the evaluation of road detection: completeness (Equation (4)), correctness (Equation (5)), and quality (Equation (6)) [34,35]:

$$Completeness = \frac{\text{length of matched reference}}{\text{length of reference}} = \frac{TP}{TP + FN} \tag{4}$$

$$Correctness = \frac{\text{length of matched extraction}}{\text{length of extraction}} = \frac{TP}{TP + FP} \tag{5}$$

$$Quality = \frac{\text{length of matched extraction}}{\text{length of extracted + unmatched reference}} = \frac{TP}{TP + FP + FN} \tag{6}$$

where *TP* (true positive) represents the length of the curb detected that matches the reference roadside, *FP* (false positive) represents the length of the detected curbs that do not match the ground truth, and *FN* (false negative) represents the total length of the undetected curbs that exist in the ground truth.

The proposed method identified 525.1 m as curbs from the Test Site 1 data; 35.8 m of that represent FP caused by low vegetation and elements with geometry similar to those of curbs, such as isolated stones, car bottoms, and stair steps; 489.3 m of the detected curb matched the ground truth curb, and 30.2 m belonging to the ground truth were not detected due to the occlusion of the curb by low vegetation or the existence of other elements, such as pedestrians or bins, over the curb (Figure 11a,b). In these cases, is not possible to achieve detection because of the change in the curb geometry. The precision indices of our method for Test Site 1 are summarized in Table 2. It can be seen that completeness and correctness are both above 90% (94.2% and 93.2%, respectively) and the quality value is 88.11%.
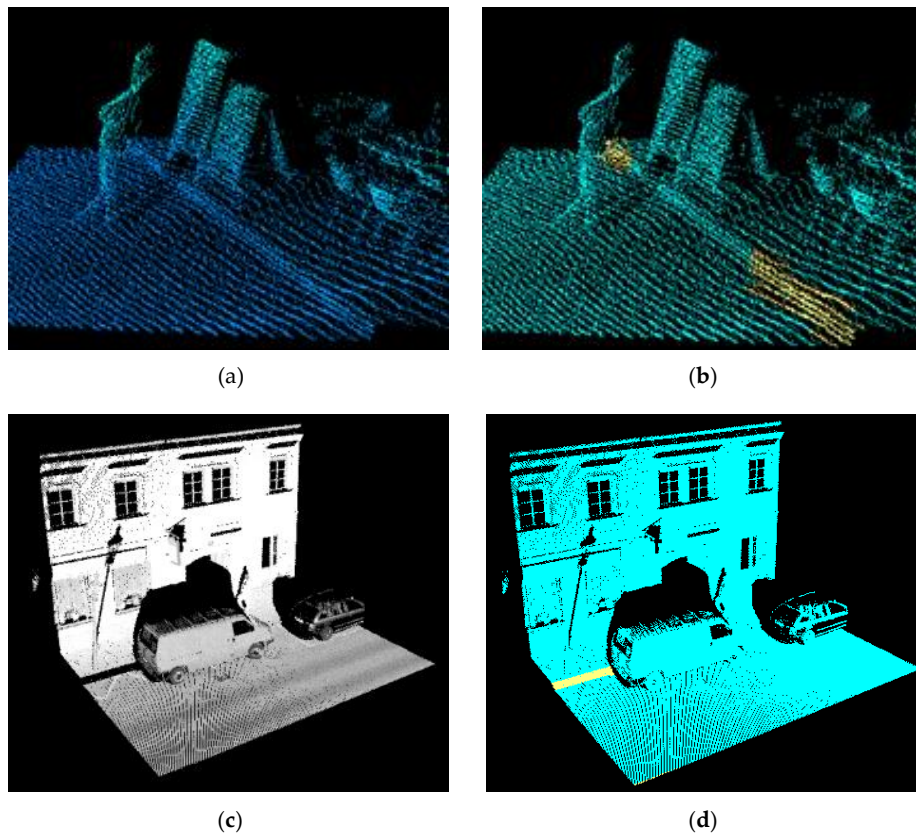

(a) (b)

(c) (d)

Figure 11. Two cases of false negatives at Test Site 1 and 2. The curb is not detected due to an element over it (**a**,**b**). The short curb between cars is removed in the morphological operation procedure (**c**,**d**).

Table 2. Accuracy of the detection method at Test Site 1.

Test Site 1	Data Present Curbs
Algorithm detected (AD)	525.1 m
User detected (UD)	519.5 m
False positive (FP)	35.8 m
False negative (FN)	30.2 m
True positive (TP = AD − FP)	489.3 m
Evaluation indices	
Completeness	94.2%
Correctness	93.2%
Quality	88.11%

4.2. Test Site 2: Curb Representation and Accuracy Evaluation

Test Site 2 corresponds to a 200-m street section in Horn, a city in the northeast of Austria (Figure 12). This is a typical urban area that has a road with structured road boundaries in the form of curbs and ramps at crosswalks and garages. One side of this street is reserved for parking, which creates shadows in the point cloud, making the detection of curbs challenging. The point cloud corresponding to Test Site 2 consisted of more than 11.5 million points. The values of the parameters used for Test Site 2 are listed in Table 3. Pixel size was fixed at 5 cm × 5 cm because the mean distance between consecutive scan lines is around 1 cm in this test site. Regarding the height difference, Hmin was set to 0.10 m and Hmax to 0.20 m. Minimum point density was set to 20 points per pixel. As in Test Site 1, roughness and neighborhood size were fixed at 2.5 and 12 cm, respectively.
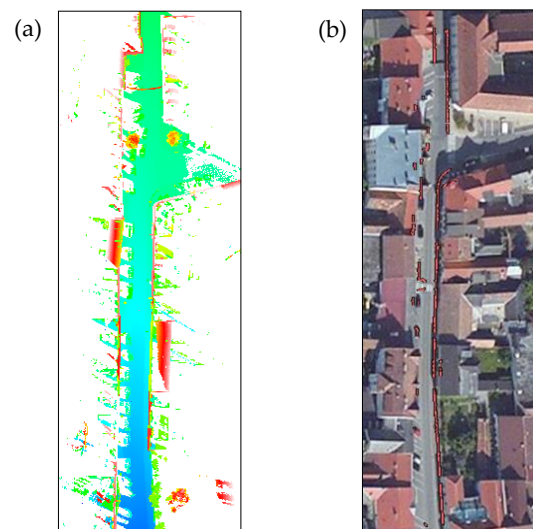

Figure 12. Point cloud measured by a RIEGL sensor used as Test Site 2 (**a**). Ortho-image of the studied area overlapped by segmented curbs (**b**).

Table 3. Algorithm settings used at Test Site 2.

Pixel Size	5 cm × 5 cm
Δh (Hmin and Hmax)	0.1 < nDSM < 0.20
Point density (Dmin)	20 points/pixel
Roughness (Redges)	2.5
Roughness neighborhood	12 cm

The outcome of the proposed method is shown in detail in Figure 13. The first row (detail A) represents a section of the street in which there is a curved curb and a sidewalk occluded by the shadow of a parked car. The proposed method correctly determines the curved section, but it is not possible to determine the curb in the occluded sidewalk due to the lack of information. Detail B (second row) is focused on a zebra crossing; in this road section, the method correctly determined the existing curbs but not the ramps to the zebra crossing due to the different height of these elements.
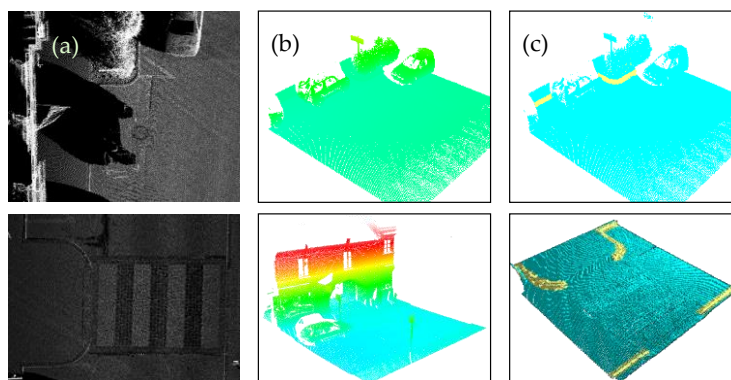

Figure 13. Details of Test Site 2. For each detail, a plan view of the intensity Lidar data (**a**), the original point cloud (**b**), and the result of the proposed method are shown with the curbs in yellow and the background in blue (**c**).

As in Test Site 1, ground truth was created by digitizing the observed curbs from the point cloud to evaluate the accuracy of the curb segmentation. The ground truth has a length of more than 220 m. The completeness, correctness, and quality indices were obtained by comparing the segmented curbs with the ground truth. The proposed method labeled 231.61 m as curbs from the Test Site 2 data. Of that, 21.4 m represent false positives caused by elements with geometry similar to curbs, such as car bottoms and stair steps (Figure 14). In total, 14.55 m of curbs present in the reference data were not detected by the proposed method due to the short distance existing between parked cars or other elements over the curbs (see Figure 11c,d). More than 210.21 m of the detected curb matched the ground truth curb. The parameters used to measure our method's accuracy are shown in Table 4. The proposed method achieved a completeness of 93.52%, a correctness of 90.76%, and a quality of 85.42%.

Table 4. Accuracy of the detection method at Test Site 2.

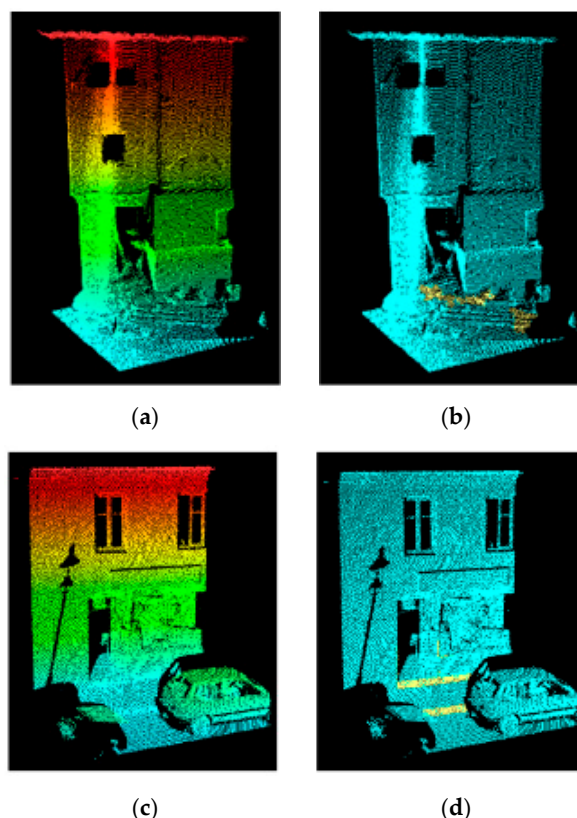| Test Site 2	Data Present Curbs
Algorithm detected (AD)	231.61 m
User detected (UD)	224.76 m
False positive (FP)	21.4 m
False negative (FN)	14.55 m
True positive (TP = AD − FP)	210.21 m
Evaluation indices	
Completeness	93.52%
Correctness	90.76%
Quality	85.42%

Figure 14. Two cases of false positives at Test Site 2. In both cases some elements with similar geometry to curbs, such as stair steps (**a** and **b**) or benches (**c** and **d**), are wrongly determined.

4.3. Parameters' Sensibility Analysis

The sensibility of each parameter has been analyzed in order to establish the range of values that every parameter can take without affecting the result of the procedure (Table 5). Table 6 shows the values of three of the variables used to estimate the accuracy when those parameters are outside of the range showed in Table 5. Some deviations from the optimum results can be appreciated.

Regarding the pixel size, it has been concluded that the best quality rates are achieved with pixel sizes between 4 and 5 times the mean distance between consecutive scan lines. Concerning height difference, the optimum Hmin and Hmax values depend on the type of curb present in the studied area. It can be summarized that Hmin can take values between 5 cm and 10 cm and Hmax between 15 cm and 20 cm. It is advisable to set minimum point density at between 15 and 25 points/pixel in order not to consider points in flat surfaces, mainly roads and sidewalks, as curbs. With regards to roughness, as can be seen in Figure 7, roughness values can take values between 2 and 2.5 with neighborhood sizes between 10 and 20 cm without affecting the final extraction quality.

Table 5. Range of values for every considered parameter and quantitative analysis for different parameter configurations.

Considered Parameters	Parameter Ranges
Pixel size	4–5 times the distance between consecutive scans
Δh (Hmin and Hmax)	[0.05; 0.10] m < nDSM < [0.15; 0.20] m
Minimum point density (Dmin)	[15; 25] points/pixel
Roughness (Redges)	[2; 2.5] roughness
Roughness neighborhood	[10; 20] cm

Table 6. Accuracy of the method for parameter values outside of the optimum range.

Considered Parameters	Completeness	Correctness	Quality
Pixel size 1 time distance between consecutive scans	67.3%	97.2%	66.03%
Pixel size 5 times distance	94.2%	93.2%	88.1%
Pixel size 10 times distance	93.3%	88.3%	83.1%
Δh (Hmin and Hmax) = [0.02; 0.17] m	98.1%	71.8%	70.8%
Δh = [0.07; 0.17] m	94.2%	93.2%	88.1%
Δh = [0.07; 0.35] m	97.1%	74.3%	72.7%
Minimum point density (Dmin) = 5	42.3%	95.7%	41.5%
Minimum point density (Dmin) = 20	94.2%	93.2%	88.1%
Minimum point density (Dmin) = 50	98.1	77.3%	76.11%

5. Conclusions

In this paper, a method to detect curbs from MLS cloud data was presented. This method involves the rasterization of the point cloud as a previous step to apply different image processing techniques such as thresholding and an opening morphological operation to determine the location of curbs in the image.

The method was tested in two datasets measured by different MLS sensors, both corresponding to urban environments. The results obtained show completeness and correctness indices higher than 90% and a quality value around 85% in both test sites. From these results, it is possible to conclude that the proposed method can be useful: (1) for curb detection in straight and curved road sections and (2) for MLS data and stereo vision point clouds, due to its independence with the scanning geometry. However, it is still difficult to deal with occluded curbs in shadowed areas and false positives caused by elements with similar properties to the curbs. In the near future, other features will be incorporated to enhance this method through decreasing the false positives rate, better curb edge detection, and estimating the location of road boundaries when they are occluded in the point cloud.

Author Contributions: Borja Rodríguez-Cuenca implemented the algorithms. Silverio García-Cortés proposed the methodology and supervised the algorithms. Celestino Ordóñez analyzed the results. María C. Alonso supervised the whole paper.

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

References

1. Wack, R.; Wimmer, A. Digital terrain models from airborne laserscanner data-a grid based approach. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2002**, *34*, 293–296.
2. Rebecca, O.; Gold, C.; Kidner, D. 3d city modelling from LiDAR data. In *Advances in 3d Geoinformation Systems*; Springer: Berlin, Germany, 2008; pp. 161–175.
3. Priestnall, G.; Jaafar, J.; Duncan, A. Extracting urban features from LiDAR digital surface models. *Comput. Environ. Urban Syst.* **2000**, *24*, 65–78. [CrossRef]
4. Schmidt, R.; Weisser, H.; Schulenberg, P.; Goellinger, H. Autonomous driving on vehicle test tracks: Overview, implementation and results. In Proceedings of the IEEE Intelligent Vehicles Symposium, Dearborn, MI, USA, 3–5 October 2000; pp. 152–155.
5. Urmson, C.; Anhalt, J.; Bagnell, D.; Baker, C.; Bittner, R.; Clark, M.; Dolan, J.; Duggins, D.; Galatali, T.; Geyer, C. Autonomous driving in urban environments: Boss and the urban challenge. *J. Field Robot.* **2008**, *25*, 425–466. [CrossRef]
6. Ding, X.; Kang, W.; Cui, J.; Ao, L. Automatic extraction of road network from aerial images. In Proceedings of the 1st International Symposium on Systems and Control in Aerospace and Astronautics, Harbin, China, 19–21 January 2006.

7.   Cheng, X.-J.; Zhang, H.-F.; Xie, R. Study on 3d laser scanning modeling method for large-scale history building. In Proceedings of the International Conference on Computer Application and System Modeling (ICCASM), Taiyuan, China, 22–24 October 2010; pp. V7-573–V7-577.

8.   Gonzalez-Aguilera, D.; Muoz, A.; Lahoz, J.; Herrero, J.; Corchón, M.; Garcia, E. Recording and modeling paleolithic caves through laser scanning. In Proceedings of the International Conference on Advanced Geographic Information Systems &, Web Services, Cancun, Mexico, 1–7 February 2009; pp. 19–26.

9.   Argüelles-Fraga, R.; Ordóñez, C.; García-Cortés, S.; Roca-Pardiñas, J. Measurement planning for circular cross-section tunnels using terrestrial laser scanning. *Autom. Constr.* **2013**, *31*, 1–9. [CrossRef]

10.   Cabo, C.; Ordoñez, C.; García-Cortés, S.; Martínez, J. An algorithm for automatic detection of pole-like street furniture objects from mobile laser scanner point clouds. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 47–56. [CrossRef]

11.   Park, H.; Lim, S.; Trinder, J.; Turner, R. 3d surface reconstruction of terrestrial laser scanner data for forestry. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Honolulu, HI, USA, 25–30 July 2010; pp. 4366–4369.

12.   Rutzinger, M.; Elberink, S.O.; Pu, S.; Vosselman, G. Automatic extraction of vertical walls from mobile and airborne laser scanning data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2009**, *38*, W8.

13.   Rutzinger, M.; Höfle, B.; Oude Elberink, S.; Vosselman, G. Feasibility of facade footprint extraction from mobile laser scanning data. *Photogramm. Fernerkund. Geoinform.* **2011**, *2011*, 97–107. [CrossRef]

14.   Hammoudi, K.; Dornaika, F.; Paparoditis, N. Extracting building footprints from 3d point clouds using terrestrial laser scanning at street level. *ISPRS/CMRT09* **2009**, *38*, 65–70.

15.   Yongtao, Y.; Li, J.; Haiyan, G.; Cheng, W.; Jun, Y. Semiautomated extraction of street light poles from mobile LiDAR point-clouds. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 1374–1386.

16.   Rodríguez-Cuenca, B.; García-Cortés, S.; Ordóñez, C.; Alonso, M.C. Automatic detection and classification of pole-like objects in urban point cloud data using an anomaly detection algorithm. *Remote Sens.* **2015**, *7*, 12680–12703. [CrossRef]

17.   Douillard, B.; Underwood, J.; Vlaskine, V.; Quadros, A.; Singh, S. A pipeline for the segmentation and classification of 3d point clouds. In *Experimental Robotics*; Springer: Berlin, Germany, 2014; pp. 585–600.

18.   Ioannou, Y.; Taati, B.; Harrap, R.; Greenspan, M. Difference of normals as a multi-scale operator in unorganized point clouds. In Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), Zurich, Switzerland, 13–15 October 2012; pp. 501–508.

19.   Yang, B.; Fang, L.; Li, J. Semi-automated extraction and delineation of 3d roads of street scene from mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* **2013**, *79*, 80–93. [CrossRef]

20.   Hu, H.; Munoz, D.; Bagnell, J.A.; Hebert, M. Efficient 3-d scene analysis from streaming data. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 2297–2304.

21.   Ruyi, J.; Reinhard, K.; Tobi, V.; Shigang, W. Lane detection and tracking using a new lane model and distance transform. *Mach. Vis. Appl.* **2011**, *22*, 721–737. [CrossRef]

22.   Labayrade, R.; Douret, J.; Aubert, D. A multi-model lane detector that handles road singularities. In Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference, Toronto, ON, Canada, 17–20 September 2006; pp. 1143–1148.

23.   Zhao, G.; Yuan, J. Curb detection and tracking using 3d-LiDAR scanner. In Proceedings of the 19th IEEE International Conference on Image Processing (ICIP), Orlando, FL, USA, 30 September–3 October 2012; pp. 437–440.

24.   Weiss, T.; Dietmayer, K. Automatic detection of traffic infrastructure objects for the rapid generation of detailed digital maps using laser scanners. In Proceedings of the IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 13–15 June 2007; pp. 1271–1277.

25.   Belton, D.; Bae, K.-H. Automating post-processing of terrestrial laser scanning point clouds for road feature surveys. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2010**, *38*, 74–79.

26.   Hervieu, A.; Soheilian, B. Semi-automatic road/pavement modeling using mobile laser scanning. *City ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *2*. [CrossRef]

27.   Kumar, P.; McElhinney, C.P.; Lewis, P.; McCarthy, T. An automated algorithm for extracting road edges from terrestrial mobile LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2013**, *85*, 44–55. [CrossRef]

28. Serna, A.; Marcotegui, B. Urban accessibility diagnosis from mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2013**, *84*, 23–32. [CrossRef]
29. Trident Trimble Software. Available online: http://kmcgeo.com/Products/Trident/Trident_Feature_Table. pdf (accessed on 5 March 2016).
30. Lehtomäki, M.; Jaakkola, A.; Hyyppä, J.; Kukko, A.; Kaartinen, H. Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. *Remote Sens.* **2010**, *2*, 641–664. [CrossRef]
31. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*; Prentice Hall: Englewood Cliff, NJ, USA, 2002.
32. Cloudcompare Wiki Main Page. Available online: http://www.cloudcompare.org/doc/wiki/index.php? title=Main_Page (accessed on 17 December 2015).
33. Mongus, D.; Lukač, N.; Žalik, B. Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 145–156. [CrossRef]
34. Heipke, C.; Mayer, H.; Wiedemann, C.; Jamet, O. Evaluation of automatic road extraction. *Int. Arch. Photogramm. Remote Sens.* **1997**, *32*, 151–160.
35. Hu, J.; Razdan, A.; Femiani, J.C.; Cui, M.; Wonka, P. Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 4144–4157. [CrossRef]