Detection and reconstruction of static vehicle-related ground occlusions in point clouds from mobile laser scanning

Liu, Zhenyu; van Oosterom, Peter; Balado, Jesús; Swart, Arjen; Beers, Bart

DOI
10.1016/j.autcon.2022.104461

Publication date
2022

Document Version
Final published version

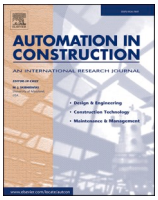
Published in
Automation in Construction

Automation in Construction 141 (2022) 104461

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon



Detection and reconstruction of static vehicle-related ground occlusions in point clouds from mobile laser scanning

Zhenyu Liu [a,*], Peter van Oosterom [a], Jesús Balado [a,b], Arjen Swart [c], Bart Beers [c]

[a] Delft University of Technology, Faculty of Architecture and the Built Environment, GIS Technology Section, 2628 BL Delft, the Netherlands
[b] Universidade de Vigo, CINTECX, GeoTECH, 36310 Vigo, Spain
[c] CycloMedia Technology B.V., Waardenburg, the Netherlands

ARTICLE INFO

Keywords:
Mobile laser scanning
Point cloud
Vehicle-related occlusion
Ground occlusion
Occlusion detection
Occlusion reconstruction

ABSTRACT

Vehicle-related ground occlusion is a common problem in MLS data. This study aims to design a detection and reconstruction method of static vehicle-related ground occlusion for MLS data. Ground extraction and vehicle segmentation are performed on the input point cloud data in advance. Then an α-shape boundary based on the prior vehicle geometry is designed to split non-ground empty area and ground occlusions. The occlusion is detected and matched with its corresponding vehicle using the relative position between them. This relative position relation and the height difference are used to detect the curb direction as the local road direction. Finally, the occlusions are reconstructed using two different methods: (1) a cell-based linear interpolation and (2) a point-based mathematical morphology. The methodology is tested by original scanned data and multi-temporal evaluation data captured from a residential area in Delft, the Netherlands with vehicle-mounted LiDAR sensors. The result shows that all occlusions cause by vehicles are successfully detected and the curb (road) direction is correctly extracted in most of the occluded areas. Both reconstructed results can visually integrate the original scanned data and recover the curb structure. The reconstruction errors of the linear interpolation method are 0.045 m in the z-axis direction and 0.051 m in total and the reconstruction errors of mathematical morphology are 0.048 m in the z-axis direction and 0.052 m in total.

1. Introduction

Mobile Laser Scanning (MLS) is an efficient and accurate approach of spatial data acquisition, also have some unignorable limitations, and one of its most common problems is occlusion [1]. The occlusion means that the capture of spatial data is incomplete. The existence of occlusion may render the point clouds data unusable because it can result in negative impacts on the entire subsequent point clouds data processing workflow, including classification, segmentation, and registration. The occlusion makes the dimensions, volume, and topology of the object in the point clouds data deviate from the real situation. In generic point clouds use, accurate geometric and topological information is often an important prerequisite to ensure correct results. Hence additional labor and time costs are needed to detect and reconstruct occlusion before or during data processing [2,3].

The distribution characteristics of occlusions are strongly affected by the point clouds data acquisition methods. For Terrestrial Laser Scanning (TLS), many point clouds data on flat roof surfaces usually cannot be captured. For Airborne laser scanning (ALS), building facades often have a lot of data missing, especially when there is an overhanging structure on the roof. As for MLS, in addition to occlusions in the flat roof (Fig. 1. (a)) and rear part of the buildings (Fig. 1. (b)) [4], the vehicle-related occlusion is also one of the most common issues (Fig. 1. (c)). Due to this reason, many occlusion reconstruction methods designed for ALS and TLS directly applied to MLS data may be less effective. For example, TLS can collect data of the same environment in multiple locations. The occlusion can be removed by merging these multi-position data [5]. But this method has some limitations when it is applied to MLS. Because of the data acquisition efficiency and cost, most MLS is based on the vehicle in practice. This means the movement trajectory of MLS has to be subject to the constraints of the road network, the traffic regulations, and the prevailing traffic conditions [2]. In addition, the occlusion in MLS data has some other characteristics. Due to moving sensor position, the occlusion by narrow objects, such as light poles, and traffic signs, is not obvious, while bigger objects (e.g., vehicles) can cause some true occlusions, but there is usually a transition zone close to the

* Corresponding author.
E-mail address: Z.LIU-46@student.tudelft.nl (Z. Liu).

https://doi.org/10.1016/j.autcon.2022.104461
Received 17 February 2022; Received in revised form 7 April 2022; Accepted 24 June 2022
Available online 3 July 2022

boundary of occlusion area (Fig. 1(c)). And the occlusion caused by other vehicles parked on the side of the road could affect the data integrity of driveways, road curbs, and sidewalks at the same time.

Among all the occlusion problems in MLS data, the vehicle-related ground occlusion deserves special attention. The complete ground surface, including driveways, road curbs, and sidewalks, plays an important role in the implementation of navigation, positioning, and other applications of point clouds. However, when collecting point cloud data, many objects located between the MLS sensor and the ground may cause ground occlusion, which can lead to many usages being unappropriate. The vehicle is usually the closest object to the sensor among all the potential occluding objects and it also has a large surface or volume, which may result in serious ground occlusions. Therefore, ground occlusion caused by vehicles has become one of the major bottlenecks to the potential uses of MLS data.

The aim of this paper is to design a detection and reconstruction method of static vehicle-related ground occlusion for MLS data. Our proposed occlusion detection method uses an α-shape based on prior vehicle geometry knowledge to fit the actual ground boundary and then uses the relative position between each occlusion with its corresponding vehicle to detect and match them. In the occlusion reconstruction phase, the curb direction is extracted to approximate the local road direction. Then, the cell-based linear interpolation and the point-based mathematical morphology respectively are used and compared with the evaluation data which is the second scan of the same area to reconstruct the occlusion along the road direction.

The paper is organized as follows: Section 2 reviews previous related research, including occlusion detection, occlusion reconstruction, and the vehicle-related occlusion issues. Section 3 presents our proposed methods. Then Section 4 describes the implementation of the methods and evaluate the results. Section 5 is devoted to discussing the performance of proposed methods. Finally, Section 6 gives conclusions of this work.

2. Related work

Occlusion detection and reconstruction in point clouds is one of the critical topics in remote sensing, SLAM, and other research fields, so many scholars have carried out relevant studies. This section first reviews the recent literature on occlusion detection (Section 2.1), occlusion reconstruction (Section 2.2), and vehicle-related occlusion (Section 2.3).

2.1. Occlusion detection

Some studies first convert 3D point clouds into 2D images by projection or depth maps and then use a binary mask to extract the occlusion areas in the image [6]. The problem with the raster-based approach is that it is difficult to determine the appropriate cell size (i.e., image resolution). If the cell size is too large, detection methods can result in inaccurate boundaries of the detected occlusion area. If the cell size is too small, detection methods may detect false occlusions in the sparse point clouds area and significantly increase the cost of computing [7]. The above problems are especially evident when dealing with surfaces with large areas and uneven density, such as the ground surface. The spatial relationship between foreground objects and their shadows (i.e., visibility analysis) is also a common method to determine the occlusion area [1]. But it is more challenging to match the foreground object with its occluded area in MLS data. The shape of the occluded area and the foreground object's shadow are not necessarily the same in the MLS data because not all foreground objects generate occlusion areas. In addition, the geometry and intensity consistency [5] and ray-tracing [8] methods are also used to detect the occlusion in point clouds data.

2.2. Occlusion reconstruction

A common solution is to remove occlusion through multi-data fusion. The first way is merging the multi-temporal data from the same source [9]. It can remove the dynamic occlusion in the data, which is caused by temporary objects such as pedestrians and moving vehicles but is not applicable to the static occlusion caused by buildings, road signs, and trees. Another way of multi-data fusion is to combine the advantages of data from different sources. Because the occlusion in MLS usually appears on the top surfaces and rear facades of the buildings, the ALS data and satellite images can be used to fill the missing parts on the top surfaces [10] and the TLS data can rebuild the occlusions on rear facades. However, the multi-source point clouds data fusion needs well-performed registration algorithm support, otherwise, it cannot eliminate differences in spatial references, observation angles, and data quality between different datasets [11]. Some studies combine these two kinds of multi-data fusion methods [12]. The advantage of multi-data fusion is that the data used to fill the occlusion is from the actual captured data, not from estimated or approximated data. But these methods also lead to the increase of data acquisition cost due to the additional data demand.

Many methods do not require additional data. Some approaches apply 2D image processing techniques [13,14] to fill occlusions after converting 3D point clouds to 2D images, usually depth maps [15]. However, depth maps also have additional disadvantages that hinder occlusion reconstruction, such as the inability to record multiple return points along the ray and limited field of view [5]. Some studies are based on the geometric characteristics of point clouds objects. Friedman and Stamos [1] apply Repeated Pattern Detection and Completion (RPDC) to divide building facades into several basic blocks as templates to fill in the occlusion areas. Some papers focus on mathematical morphology. Serna and Marcotegui [7] use morphological interpolation to fill holes in DEM data but it is still base on raster data. Recently, many papers have also tried to use deep learning to solve the occlusion problem. Yuan et al.

(a) roof occlusion (b) rear facade occlusion (c) vehicle-related occlusion

Fig. 1. Typical occlusions in MLS data (Data source: Clyclomedia).

[16] raise a Point Completion Network (PCN) to complete the shape of point cloud objects. Barazzetti [17] designs shallow feedforward neural networks to recover the point cloud occlusions. Czerniawski et al. [18] present a hierarchical deep variational autoencoder to filling-in the missing point clouds data. These reconstruction methods based on deep learning have achieved many encouraging results. However, due to the high computational cost of deep learning methods, most of the previous studies focused on the reconstruction of vehicles, building facades, and other small objects [19], but the reconstruction of large-scale objects, like road surface, is not much involved.

2.3. Vehicle-related occlusion

MLS usually contains three types of vehicle-related occlusions: (1) static ground occlusions caused by parked vehicles, (2) temporary ground occlusions caused by dynamic or running vehicles, and (3) vehicles' self-occlusions (i.e., the absence of point cloud data in the car body). Many studies focus on detecting and reconstructing vehicle self-occlusion in point clouds data [20,3]. Some researchers have discussed the influence of dynamic occlusion caused by vehicles on the localization ability of point cloud data [21,22]. Balado, González, et al. [2] discusses the matching method between common environmental objects such as vehicles and ground occlusions caused by them. Compared with the other two kinds of vehicle related occlusions, the research on the vehicle-related ground occlusions is not deep enough. Most studies use generalized dynamic and static ground occlusion processing methods to solve this problem while lacking a specialized approach for vehicle-related ground occlusion.

3. Methodology

The methodology is divided into three main phases: (1) Pre-processing phase prepares the data, such as classifying ground surface and vehicles, for the next two phases from the raw MLS data (Section 3.1). (2) Occlusion detection phase determines the ground boundary, detects each ground occlusion, and matches it with its corresponding vehicle (Section 3.2). (3) Occlusion reconstruction phase first obtains the local curb direction and then uses linear interpolation and mathematical morphology to fill occlusions respectively (Section 3.3). The whole workflow of the methodology is shown in Fig. 2.

3.1. Pre-processing

The pre-processing phase includes two operations: ground surface extraction and vehicle segmentation. These two operations are not the key research problems of this study but are essential for the subsequent phases. To evaluate the robustness of the proposed methods, the expected outcome of the pre-processing phase in this study is not high-quality segmented ground and vehicle objects, but some low-quality segmentation objects are consciously added into the result.

3.1.1. Ground surface extraction

The first pre-processing operation is to extract the ground surface from the whole MLS data set. Since this research only concerns ground occlusions, the detection and reconstruction methods can be applied to the ground surface directly, instead of the whole MLS data set, which can obviously reduce the computational cost. In this research, a mean elevation map with a grid resolution of res_{mem} (0.2 m) is used to extract the ground surface [23,24]. For each occupied grid cell in the mean elevation map, its elevation is the average height of all points located in this cell. Then the elevation of every cell is compared with the elevation of each occupied neighbor cell. If their height difference is less than the threshold $thr_{ground\Delta z}$ (0.03 m), this pair of neighbor cells are connected. Because the mean elevation map is a 2.5D structure, only 8-connectivity needs to be considered for each cell. After traversing all cells, all connected cells can form a cell set, corresponding to a horizontal plane or a flatly changed surface. The largest set of connected cells corresponds to the ground surface.

The extracted ground surface may contain some outliers. The cause of this problem can be noise in the original MLS data set, or remnants of point cloud objects such as buildings, grass, vehicles, etc. These outliers may change the geometric characteristics of the local ground, such as increasing roughness and height difference of the local ground surface. These abnormal geometric characteristics may interfere with curb detection. This paper provides a solution to this problem in Section 3.3.1.

3.1.2. Vehicle segmentation

After removing the ground surface from the initial MLS data set, the rest is a non-ground data set. The second pre-processing operation is to segment the vehicles from this non-ground data set. The main idea of the vehicle segmentation is first to generate a 3D voxel grid in the non-ground data set with a grid resolution of res_{vg} (0.15 m) and then all occupied voxels are split into several independent objects by 26-connectivity-based region growth [25]. Next, the prior knowledge of vehicles is applied to select potential vehicle objects among all segmented objects (see Table 1). But complete non-ground data set is usually very large in space, especially if there are many very tall buildings in this area. This

Table 1
Prior knowledge of vehicles.

Prior knowledge	Min value	Max value
Length	3 m	6 m
Width	1.5 m	3 m
Height	1.3 m	2.5 m
Length-width ratio	1.5	3
Intensity	0	8000

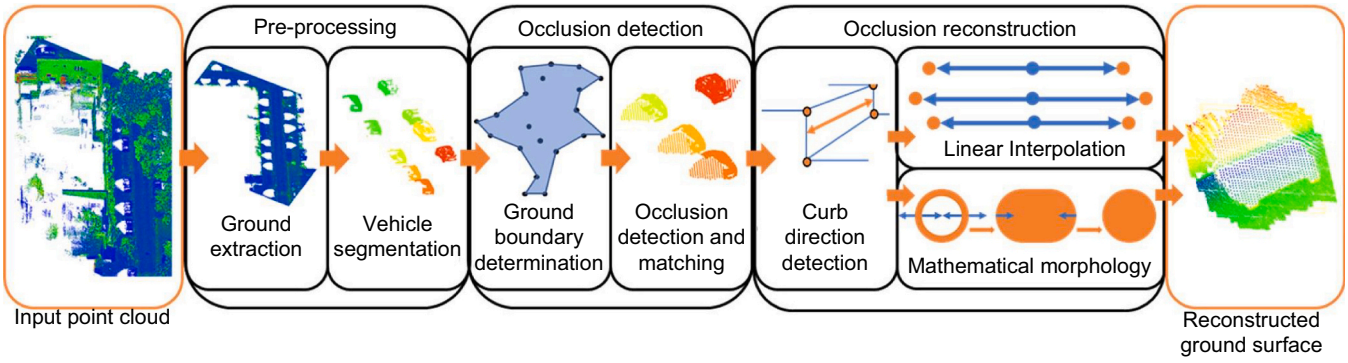


Fig. 2. Workflow of the methodology.

results in the generated voxel grid with a large number of cells, which means checking the connectivity of each cell can be very time-consuming. Therefore, we need to extract a subset of point clouds including all vehicle objects from the complete non-ground data set to reduce the calculation time. According to the prior geometry knowledge of vehicles, if we only consider the case of most private cars, excluding special vehicles such as buses and trucks, most vehicles are not higher than 2.5 m. So we make a buffer of 2.5 m from the ground up to get a 3D space. Since the ground may be a curved surface with a certain height variation, we buffer each independent ground grid cell up separately. The ground grid cells which miss the ground elevation can be roughly estimated by Nearest Neighbor Interpolation (NNI). We create a smaller 3D voxel grid only for the point clouds included in this 3D space to reduce the processing time.

The height attributes of the vehicle object are no longer needed after the segmentation operation, so the segmented vehicle objects can be simplified as 2-dimensional oriented bounding boxes using the Rotating Calipers Algorithm [26], which can easily deduce the key 2D geometric information and avoid storing all the points of vehicle objects.

The results of this segmentation method not only contain many accurate segmented vehicles but also include some vehicle objects which are not segmented very accurately, such as the segmentation results mixed with the remains of other environment objects (Fig. 3. (a)) and incomplete segmentation due to severe self-occlusion (Fig. 3. (b)). These low-quality segmentation objects generate oriented bounding boxes of vehicles with wrong sizes, positions, or orientations, which may result in obtaining the wrong curb orientation. So we can use them to verify the dependence of the proposed detection and reconstruction methods on the accuracy of pre-processing and the robustness of processing low-quality segmentation results. The relevant analysis is present in Section 4.

3.2. Occlusion detection

In the occlusion detection phase, the first task is to determinate the actual boundary of ground surface (i.e., the target area of the occlusion detection method). Then the second task is setting the center of oriented bounding box as the seed point to detect and match each occlusion with its corresponding vehicle.

3.2.1. Ground boundary determination

Before starting to apply the occlusion detection method, the actual boundary of the ground within the scanned area needs to be determined (the green boundary in Fig. 4. (b)). Only the 2D boundary is needed in this phase, so the extracted 3D ground surface can be projected on a 2D plane at the beginning. There are one or more empty areas in the minimum boundary rectangle region of the extracted ground surface. The empty area inside the actual ground boundary is the ground occlusion area, otherwise, it is a non-ground area. The occlusion detection method should only be applied to ground occlusion areas, not non-ground areas. A simple solution to determinate the boundary of the extracted ground surface (named object boundary) is drawing a circle with a fixed length radius centered on each point as the neighborhood of this point. If four quadrants of this circle all contain other points, the center point is an object's inner point, otherwise, it is an object's boundary point. But the problem is many occlusions are semi-close areas, which means they are connected to the non-ground area (Fig. 4. (a)). If directly using the object boundary (the orange boundary in Fig. 4. (b)), these semi-close occlusions are excluded from the ground areas, causing the occlusion detection method missing them.

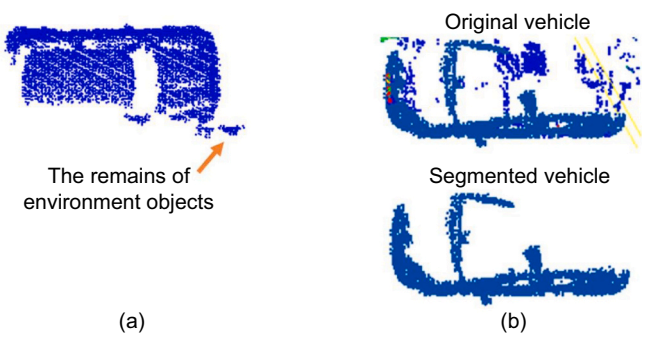


Fig. 3. Low-quality segmentation objects.

Due to the inconsistency between the actual boundary and object boundary of the ground surface, the 2D α-shape [27] is introduced to generate a new boundary that fits the actual boundary better. This new boundary can include both whole-close occlusions and semi-close occlusions. The α-shape algorithm aims to obtain the concave boundary of a point set. The algorithm first requires defining a circle of a fixed radius with a solid interior and then rolling this circle around a point set. Any point in this point set cannot enter the interior of this circle. If any pair of points in the set of points are on the boundary of the circle at the same time during the roll, an edge is generated between the pair of points. After the roll is complete, all the generated edges can form one or more polygons, called α-shapes. For the α-shape, the choice of the α value (the circle radius) is crucial. The boundary of the α-shape with a very small α value can be very similar to the object boundary which excludes the semi-close occlusions (the orange boundary in Fig. 4. (b)). If the α value is further reduced ($\alpha \rightarrow 0$ m) to less than the distance between any pair of neighbor points in the point set, the algorithm cannot construct any polygon. Conversely, if the α value continues to increase ($\alpha \rightarrow \infty$), the algorithm will end up with a convex hull rather than a concave hull of this point set. Therefore, only finding a moderate α value can generate an α-shape fitting the actual boundary well (the blue boundary in Fig. 4. (b)).

Considering the acquisition method of MLS data, the maximum width of occlusion caused by vehicles is usually less than the diagonal length of the vehicle's oriented bounding box (Fig. 5). Based on the maximum length (6 m) and maximum width (3 m) in Table 1, the maximum diagonal length $l_{maxdiag}$ is around 6.7 m. If the diameter of the circle in the α-shape algorithm is also set to $l_{maxdiag}$, it cannot enter the semi-occlusion areas when it rolls. So the moderate α value should be half of $l_{maxdiag}$, which is 3.35 m. In Fig. 15, the result shows that all semi-occlusions are successfully included in the ground area.

3.2.2. Occlusion detection and matching

The spatial topological relationship is missing in the disorderly point cloud data, so it is challenging to describe the geometric information, such as shape and position, of occlusion areas in the point cloud directly. For this issue, a 2D grid structure can be used to represent the occlusions and increase the detection efficiency. So firstly the ground surface needs to be split into a 2D ground grid with a resolution of res_{mem} (0.2 m). The cell in the ground grid which contains any point is marked as occupied cell, otherwise, it is an empty cell. The edges in the α-shape boundary also should be converted into cells in the ground grid by using Bresenham's line algorithm [28]. These boundary cells are considered occupied cells even if they are empty. Some rasterized boundary edges may have the bridge cells issue. The bridge cells are a pair of connected diagonal neighbor cells. In the ground grid, all occupied cells are considered disconnected, while empty cells are connected to each other. So the bridge cells can let the occlusion detection method connect the semi-close occlusions to the non-ground areas across the α-shape boundary. For boundary cells, a simple solution of bridge cells is making the boundary thicker in the ground grid (Fig. 6).

The next step is detecting all vehicle-related occlusions in the ground grid. Checking all empty cells should be avoided since occlusion is not the only reason that causes empty cells. Since the α-shape boundary is impossible to fit the actual boundary perfectly, some small non-ground areas, which may result in empty cells, are contained in the α-shape. Some areas far from the LiDAR sensor can also generate empty cells due

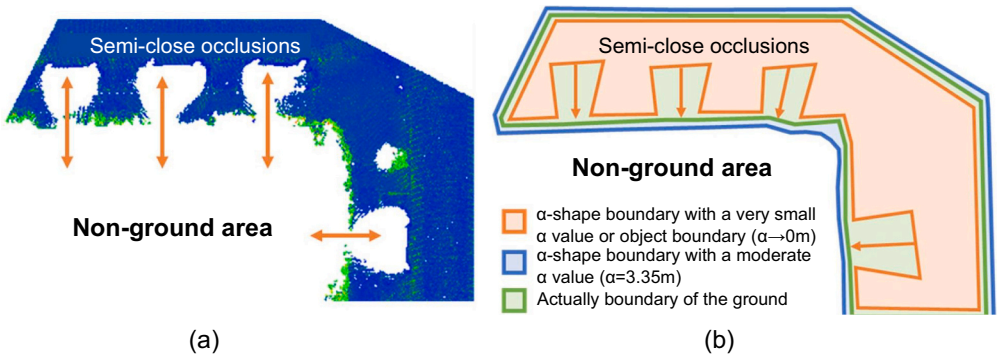


Fig. 4. Different boundaries of the ground surface with semi-close occlusions and a corner.

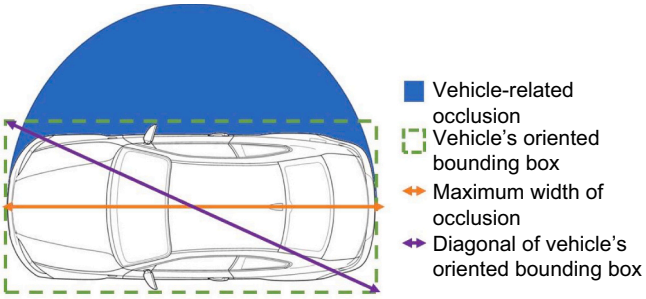


Fig. 5. The maximum width of occlusion and the diagonal length of vehicle's oriented bounding box.

to sparse point clouds, especially when the ground grid has a very small res_{mem} value. So not all empty cells can be marked as occlusion cells.

To obtain vehicle-related occlusion cells from all the empty cells, we find a seed cell (i.e., the cell must in the occlusion area) for each occlusion region and extracted all the occlusion cells using 2D spatial adjacency. Considering that the vehicle causing the occlusion is located above its corresponding occlusion area, the geometric center of this vehicle's oriented bounding box is located inside the occlusion area. Therefore, the matching relationship between the vehicle and its corresponding occlusion area can be directly established through the relative position between them. The cell which contains the geometric center of this vehicle's oriented bounding box can be set as the seed cell, which is the first cell of currently detected occlusion. Then start from the seed cell, setting it as the detecting cell and all unvisited empty neighbor cells of the detecting cell are marked as occlusion cells by using 8-connectivity. Repeat the above operation for all newly marked occlusion

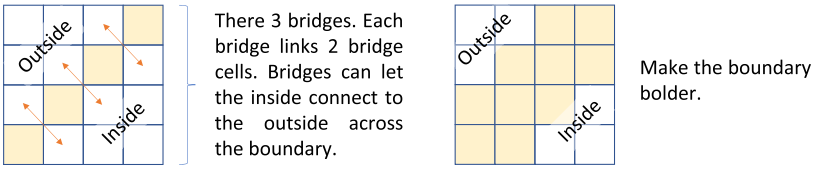


Fig. 6. The bridge cells issue of boundary cells and its solution.

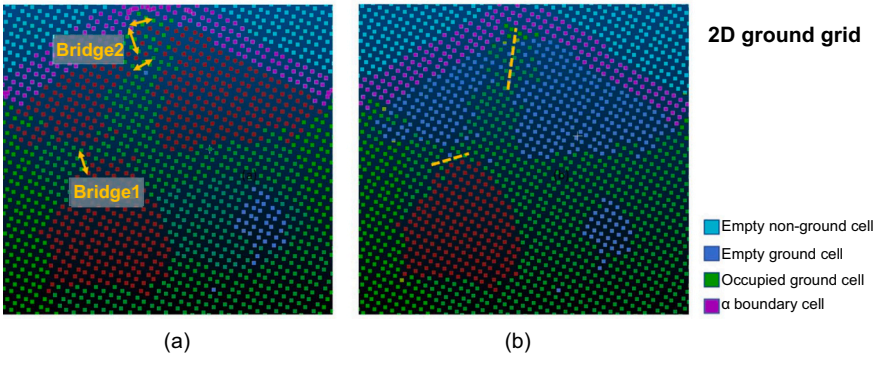


Fig. 7. Detection results before (a) and after (b) the bridge removal operation.

cells until no new unvisited empty neighbor cells appear.

The bridge cells issue also occurs during the occlusion cell detection process. The bridge can connect multiple independent empty areas into one occlusion area (Fig. 7. (a)). Therefore for each new unvisited empty neighbor cell, it should be checked if it forms a pair of bridge cells with the current detecting cell. If so, this neighbor cell should be skipped to cut off the bridge (Fig. 7. (b)). Algorithm 1 describes the method of automatically detecting bridge cell(s) from all connected neighbor cells of each detecting cell.

Algorithm 1. Bridge cell detection.

Input: Detecting_Cell C_{detect}
Output: Bridge_Cells $\{C_{bridge}\}$
// Define two global list objects $\{C_{empty\_nb1}\}$ and $\{C_{empty\_nb2}\}$.
For C_{nb1} in NeighborCellDetector(C_{detect}) **do**
  If $C_{nb1}.is\_empty()$ = True
    $\{C_{empty\_nb1}\} \leftarrow C_{nb1}$
  End_if
End_for
For $C_{detect1}$ in $\{C_{empty\_nb1}\}$ **do**
  For C_{nb2} in NeighborCellDetector($C_{detect1}$) **do**
    If $C_{nb2}.is\_empty()$ = True
      $\{C_{empty\_nb2}\} \leftarrow C_{nb2}$
    End_if
  End_for
  If $\{C_{empty\_nb1}\}$ and $\{C_{empty\_nb2}\}$ don not have any common cell
    $\{C_{bridge}\} \leftarrow C_{detect1}$
  End_if
  $\{C_{empty_{nb2}}\}.clear()$
End_for
Return $\{C_{bridge}\}$

3.3. Occlusion reconstruction

The targets of the occlusion reconstruction phase are filling the incomplete ground surface with occlusions as a continuous surface and recovering the 3D structure of the ground. In Fig. 8 top, the cross section view (perpendicular to the road direction) of the extracted 3D ground contains two main spatial characteristics: (1) There is an obvious height variation at the curb area, which is the boundary between the driveway and the sidewalk. (2) The driveway is not a horizontal plane but a curved surface, which is high in the middle (top area of the driveway), and low on both sides (the area close to the curb). On the contrary, the lengthwise section view (parallel to the road direction, see Fig. 8 bottom) is relatively flat which is easier to fit. Consequently, this study focuses on the reconstruction methods of occlusion along the road direction. The first task of this phase is the approximation of the road direction by using the curb direction (see Section 3.3.1). The second task is the reconstruction of occlusion with either (1) linear interpolation (Section 3.3.2) or (2) mathematical morphology (Section 3.3.3).

3.3.1. Curb detection

Obtaining accurate local road direction is an important prerequisite for following occlusion reconstruction steps. Due to the consistency between the curb direction and the road direction in the local region, the curb direction can be a good estimate of the road direction [29]. But the performance of previous curb detection methods depends on the curb occlusion condition. Therefore a curb detection method for occlusion environment is proposed in this step for determining the road direction.

First, a buffer area with a distance of l_{buffer} (1 m) is made around the occlusion area as its neighborhood. The buffer area needs to be converted into a mean elevation map for finding the max height difference between each map cell and its neighbor cells. If the max height difference is larger than the threshold $thr_{curb\Delta z}$ (0.03 m), this cell is marked as a potential curb cell. All connected potential curb cells form a large height difference area. The center of this area is defined as the geometric center of all the points it contains. Two curb areas can be found among all large height difference areas. But the curb is not the only reason that led to the large height difference. The remains of non-ground objects (like vehicles and bush) after segmentation and grass areas all are possible causes of this issue (Fig. 9). So the vehicle object can be used to locate these two curb areas. The oriented boundary box of the vehicle generated in the last step of Section 3.1.2 has two long edges, one is closer to the centerline of the driveway (inner edge), and another is closer to the curb (outer edge). Usually, the vehicle is not located right above the occlusion center, but slightly to the side of the road centerline. Therefore, the long edge closer to the occlusion center is the outer edge of the boundary box. Then looking for the large height difference area closest to each vertex of this edge, which is the target curb area. Finally, the local curb (road) direction can be obtained by connecting the two curb area centers. Fig. 10 shows the workflow of curb detection. In the whole curb detection process, the grid of the mean elevation map only plays an auxiliary role. The key elements of the curb direction (vertices of the oriented boundary box and centers of curb areas) are obtained directly from the point cloud. Therefore, this curb detection method can be considered as a pure point cloud operation.

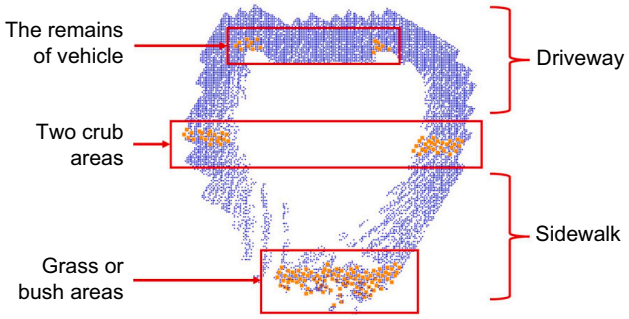


Fig. 9. The areas with large height difference in the occlusion buffer area.

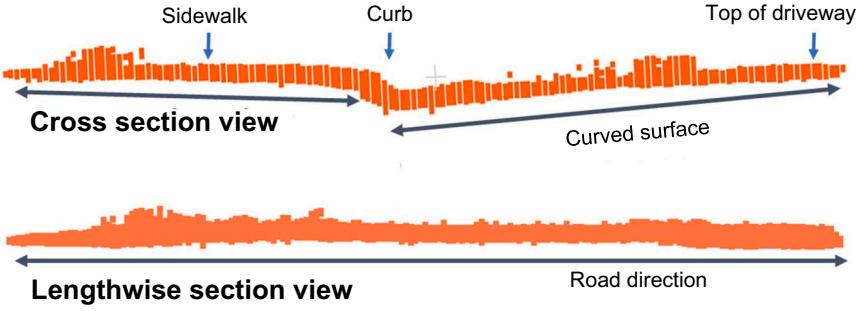


Fig. 8. The cross section view and lengthwise section view of extracted 3D ground surface.

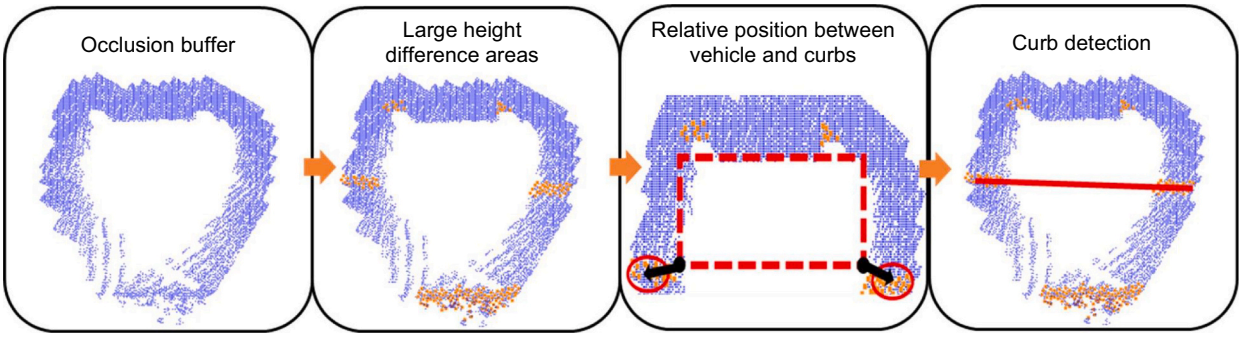


Fig. 10. Workflow of curb direction detection.

3.3.2. Linear interpolation

The curb line splits the ground into three parts: sidewalk, driveway, and curb area. The curb area in the ground grid can be detected by rasterization of the curb line with Bresenham's line algorithm. For the remaining two areas, the one that includes the center of the corresponding oriented boundary box of the vehicle is the driveway, and the other is the sidewalk.

Both sidewalks and driveways are considered 2.5D structures, which means it can be assumed that no façade is in these areas. So their reconstruction solution is similar. First a new 2D ground grid with a grid resolution of res_{grid} (alternative values are shown in Table 2) is created. For each empty cell from this new grid in these two areas, it needs to draw a parallel line of curb line passing through the 2D cell center *pt*. If the parallel line gets two intersected cells nearest to *pt* from both sides, the heights of these two intersected cells ($h_1$and h_2) can be obtained from the mean elevation map. According to linear interpolation theory, the ratio of the height difference between the current empty cell (h) and the two intersected cells ($h_1$and h_2) is equal to the ratio of its 2D distance to the two intersected cell centers ($l_1$and l_2) (Eq. 1). Finally, point *pt* is inserted at height h as a new reconstructed point.

$$h = h_1 + \frac{l_1}{l_1 + l_2} \cdot (h_2 - h_1) \tag{1}$$

The curb area is considered a 3D structure. For reconstruction of its 3D details, a height range computed by h_{max} and h_{min} is needed. Firstly, it still needs to draw a parallel line of curb line passing through *pt* and find two intersected cells nearest to *pt* from both sides. The heights of the highest points (h_{max1} and h_{max2}) and the lowest points (h_{min1} and h_{min2}) in the two intersected cells can be used to interpolate h_{max} and h_{min} respectively using Eq. 1 (Fig. 11. (a)). For each curb cell, intermediate points with z-axis interval l_{invl} (0.05 m) are vertically added between h_{max} and h_{min} at cell center *pt* as reconstructed curb points (Fig. 11. (b)). The reconstructed curb point is completely vertical in a single curb cell but may have a small angle for the entire curb surface (see Fig. 21 top).

The planar density of reconstructed points depends on the resolution of the ground gird (res_{grid}). The value of l_{invl} influences the vertical density of reconstructed curb points.

Table 2
Implementation parameters.

Parameter	Alternative value(s)
res_{mem}	0.2 m
$thr_{ground\Delta z}$	0.03 m
res_{vg}	0.15 m
α	3.35 m
res_{grid}	0.2 m/0.1 m/0.05 m
l_{invl}	0.05 m
l_{buffer}	1 m
$thr_{curb\Delta z}$	0.03 m
s_{SE}	21/41/85
l_{SEinvl}	0.4 m/0.2 m/0.1 m

3.3.3. Mathematical morphology

Mathematical morphology (MM) is a commonly used digital image processing method based on set theory and topology [30]. Recently, MM is also directly applied to 3D point cloud processing without converting it into raster data. Structuring elements (SE) and basic morphological operations (dilation, erosion, opening, and closing) applicable to point clouds have been designed [31]. In this study, a 1D linear SE and morphological closing operation are used to recover the occlusion.

The morphological closing operation is a composite operation of morphological dilation and erosion. The main idea is dilating simultaneously to the inside and outside of the buffer area along the SE direction to fill the occlusion area, and then the outer boundary of buffer area is recovered to the original shape by morphological erosion (Fig. 12).

For the 1D linear SE, three parameters (see Fig. 13) need to be considered: (1) Size s_{SE} (alternative values are shown in Table 2) is the number of points in the SE. (2) Orientation of the SE, which is the same as the road (curb) direction. (3) Interval l_{SEinvl} (alternative values are shown in Table 2) is the distance between two neighbor points in the SE, and it also decides the linear density of points in the SE. The total length of SE l_{se} is equal to l_{SEinvl} (s_{SE} – 1). It must be long enough to ensure that the morphological dilation completely fills the occlusion area without leaving any small occlusion area. Otherwise, the morphological dilation erosion will enlarge this small residual occlusion area, resulting in an incomplete reconstruction.

4. Experimental results

In this section, an original scanned data and a second scanned data set for evaluation are used to test the methodology. The C++ code was run on an AMD Ryzen 9 CPU 3.30 GHz with 16 GB RAM.

4.1. Original scanned data and evaluation data

The original scanned data for the methodology was captured by a single vehicle-mounted Velodyne HDL-32E LiDAR sensors from Cyclo-Media Technology in June 2019 in Delft which is a city in the province of South Holland, the Netherlands. The data includes 4.9 million points with 3D coordinates and intensity information. It contains a 50-m by 50-m residential area and its mean surface density is 2932.771 pts./m^2 when the search radius is 0.1 m. It includes a road with corners, sidewalks, street-facing buildings, trees, and multiple vehicles parked against the curb. These parked vehicles all result in ground occlusions (see Fig. 14).

The methodology uses the relative position relation between the vehicle and its corresponding occlusions in many steps. Therefore, the detection and reconstruction results cannot be evaluated by artificially generating occlusions in the same data, because the corresponding vehicle objects of these artificial occlusions are missing. So in this study, multi-temporal data is introduced to evaluate the results. The evaluation data was collected in the same manner in the same area in July 2020,

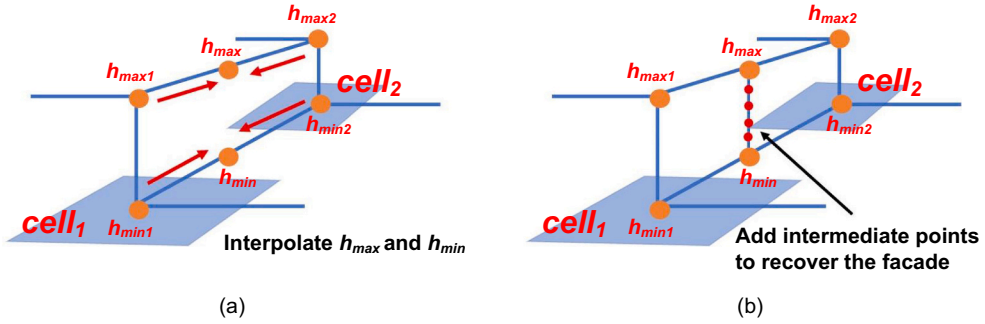


Fig. 11. Reconstruction of curb using linear interpolation.

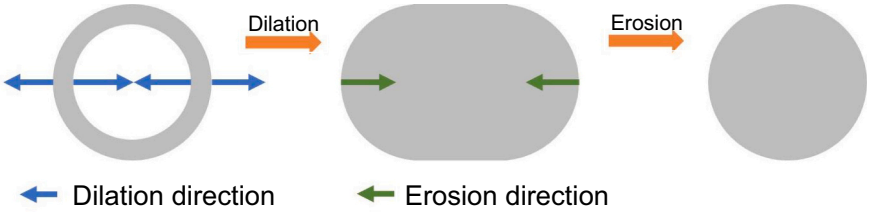


Fig. 12. Occlusion reconstruction with morphological closing.

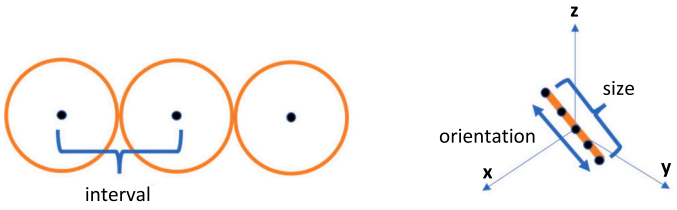


Fig. 13. Three parameters of the 1D linear structuring element.

including 5.9 million points. Its mean surface density is 3283.357 pts./m^2 when the search radius is 0.1 m. The common non-occluded ground areas of implementation and evaluation data are extracted for comparison in the software CloudCompare by using Cloud/Cloud Dist. Module to compute the point-to-surface distance. The result shows that there is almost no difference between the two data in the z-axis direction. The average absolute 3D distance is 0.003 m, max absolute 3D distance is 0.023 m, and the standard deviation is 0.005 m. Therefore the differences between the implementation and evaluation data are acceptable.

4.2. Implementation parameters

Parameters used for the original scanned data are shown in Table 2. The moderate α value is calculated based on prior vehicle geometry in Section 3.2.1. The value of res_{mem} (0.2 m) is based on previous studies [23,24]. The slope of urban roads generally does not exceed 10% but considering the measurement error and local road deformation, we extend the upper limit of slope to 15%. Within this slope limit, we set maximum allowable height differences $thr_{ground\Delta z}$ and $thr_{curb\Delta z}$ as 0.03 m. res_{vg} is set to 0.15 m to keep the balance between voxel grid resolution and computational cost. To limit the number of points in the buffer area while obtaining sufficient information about the surrounding environment, l_{buffer} is set to 1 m. Different values are set to res_{grid}, s_{SE}, and l_{SEinvl} for analyzing the reconstructed results with different densities.

4.3. Occlusion detection result

As shown in Fig. 15, all 14 vehicle-related occlusions in implantation data are successfully detected. The average time of multiple executions of the occlusion detection over the whole original scanned data is 63 s. Although the α-shape boundary includes part of non-ground empty areas, most occlusions are not connected to them due to the bridge detection algorithm, except for the 11th occlusion.

4.4. Occlusion reconstruction result

4.4.1. Curb detection

The results in Fig. 16 show that most of the curb areas can be detected correctly. In general, adjacent curbs are aligned correctly and in the same direction. The opposite curb of the road is parallel to each other. A significant change in road direction does not affect the detection results. However, the results with poor accuracy are mainly located at the boundary of the MLS data tiles (the 1st occlusion in Fig. 16) and in areas where the height changed significantly (the 7th occlusion in Fig. 16). The abnormal oriented boundary box does not significantly affect curb detection (Fig. 17).

4.4.2. Linear interpolation

The linear interpolation method is applied to reconstruct the original scanned data with different res_{grid} values, which can generate the 0.2 m/0.1 m/0.05 m grid of reconstructed points. If the reconstructed area in the original scanned data is not missing at the corresponding position in

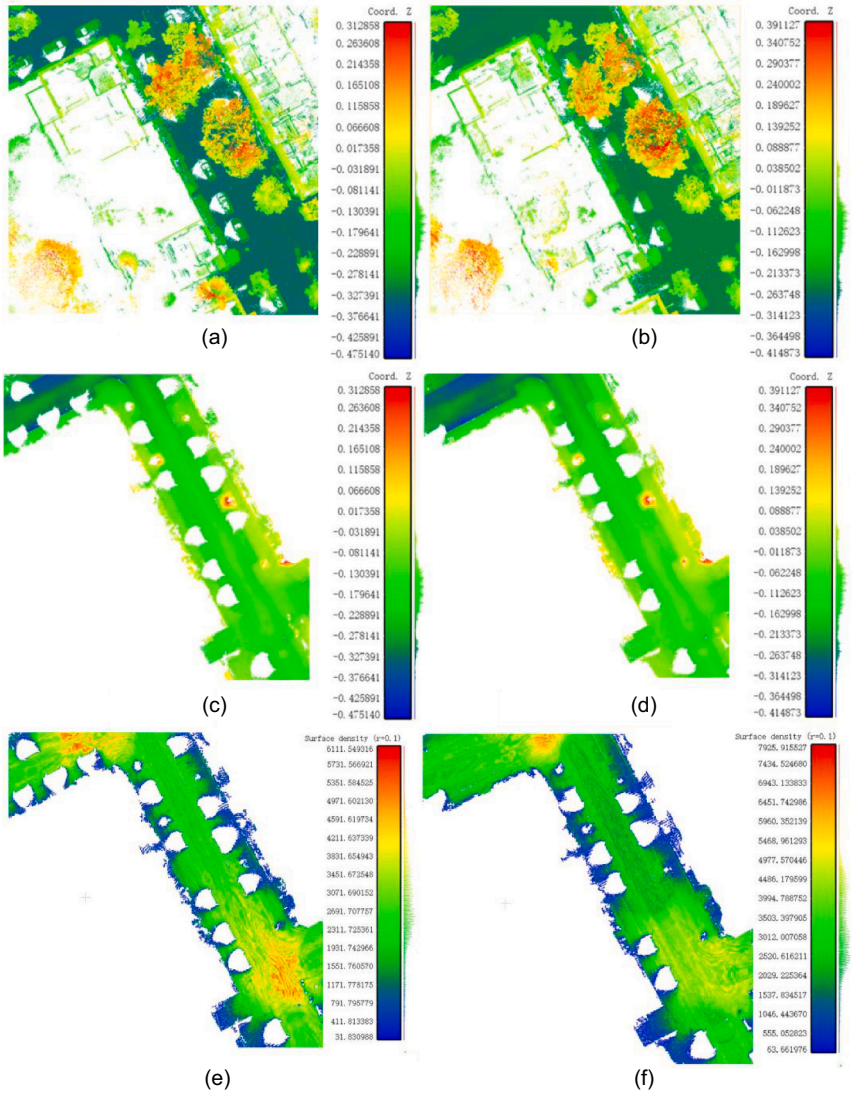


Fig. 14. Original scanned data (a), evaluation data (b), ground of original scanned data (c), and ground of evaluation data (d) rendered in height. Surface density of original scanned data (e) and surface density of evaluation data (f) when the search radius equals to 0.1 m.

Fig. 15. The result of the occlusion detection (the scene is now street-aligned by rotating clockwise the scene from Fig. 14).

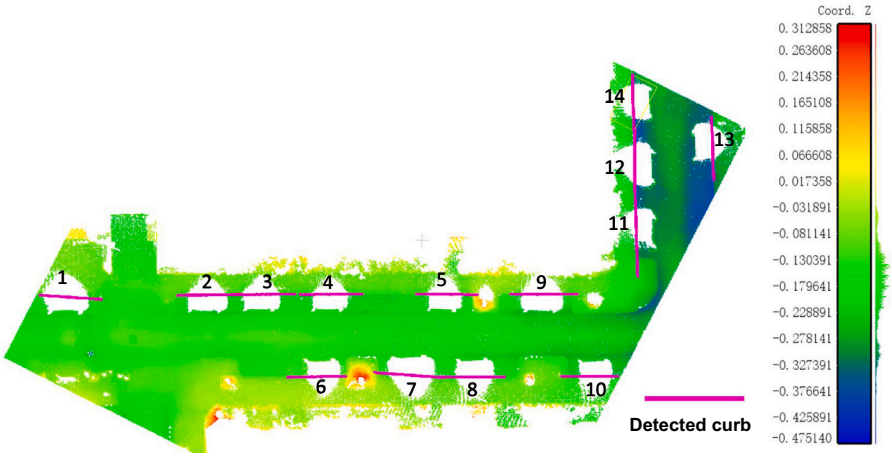


Fig. 16. Ground surface rendered in height and detected curbs.

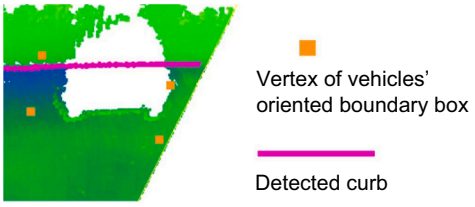


Fig. 17. The relative position between inaccurate oriented boundary box of vehicle and curb (extracted and rotated clockwise from the 14th occlusion in Fig. 16).

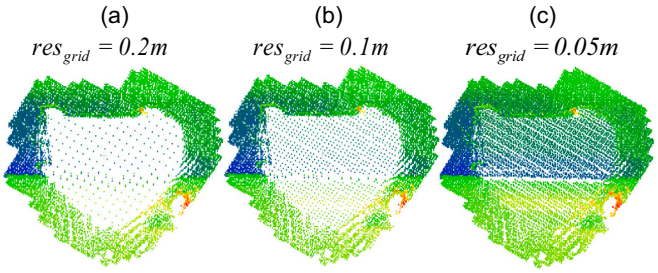


Fig. 18. Reconstructed results using linear interpolation with different res_{grid} (extracted and rotated clockwise from the 13th occlusion in Fig. 16).

Table 3
Performance of reconstructed results using the linear interpolation method.

res_{grid}(m)	0.2	0.1	0.05
Curb max abs. 3D dist. (m)	0.141	0.134	0.094
Curb avg. abs. 3D dist. (m)	0.049	0.047	0.034
Curb st.dev 3D abs. dist. (m)	0.028	0.024	0.020
Curb max z-axis abs. dist. (m)	0.129	0.133	0.089
Curb avg. z-axis abs. dist. (m)	0.042	0.038	0.026
Curb st.dev abs. z-axis dist. (m)	0.026	0.024	0.018
Non-curb max abs. 3D dist. (m)	0.126	0.096	0.105
Non-curb avg. abs. 3D dist. (m)	0.055	0.055	0.055
Non-curb st.dev abs. 3D dist. (m)	0.013	0.013	0.013
Non-curb max abs. z-axis dist. (m)	0.125	0.082	0.086
Non-curb avg. abs. z-axis dist. (m)	0.052	0.052	0.052
Non-curb st.dev abs. z-axis dist. (m)	0.013	0.012	0.013
Total max abs. 3D dist. (m)	0.141	0.134	0.105
Total avg. abs. 3D dist. (m)	0.051	0.051	0.051
Total st.dev abs. 3D dist. (m)	0.025	0.019	0.017
Total max abs. z-axis dist. (m)	0.129	0.133	0.089
Total avg. abs. z-axis dist. (m)	0.045	0.046	0.047
Total st.dev abs. z-axis dist. (m)	0.024	0.020	0.018
Reconstructed points number	8218	24,733	63,625
Time in total (sec)	<1	1	4

the evaluation data, this reconstructed area is extracted to measure its reconstruction quality in CloudCompare using Cloud/Cloud Dist. Module by computing the *abs. 3D dist.* and the *abs. z-axis dist.* The *abs. 3D dist.* is the absolute 3D distance between the reconstructed point and its nearest point in the evaluation data. The *abs. z-axis dist.* is the z component of the *abs. 3D dist.* Because this method is optimized specifically for curbs, the performance of curb areas, non-curb areas, and the total area are evaluated respectively (Table 3).

The smaller res_{grid} value results in the rising of reconstructed ground point density, and longer running time. But the computing speed is still very fast. The smaller res_{grid} value does not significantly enhance the reconstruction accuracy of non-curb areas and even has a negative effect on the total z-axis accuracy. However, it improves the reconstruction accuracy of curbs and produces a more uniform point distribution (lower standard deviation). In addition, the smaller res_{grid} can also bring a clearer curb structure and better fitting of reconstruction points to the boundary of buffer areas (Fig. 18). The distribution of points depends entirely on the shape and size of the grid.

4.4.3. Mathematical morphology

In the realization process of MM, three different combinations of l_{SEinvl} and s_{SE} are used to generate reconstruction results with different densities while keeping the l_{se} relatively unchanged (between 8 m and 8.5 m). Then, the total performance of the reconstructed results with different densities is analyzed using the same method applied to evaluate linear interpolation results (Table 4).

The execution time of MM is very long (see Table 4) because the morphological dilation generates a mass of new points. Most of these newly generated points need to be removed in the morphological

Table 4
Performance of reconstructed results using the mathematical morphology method.

l_{SEinvl}(m)	0.4	0.2	0.1
s_{SE}	21	41	85
Total max abs. 3D dist. (m)	0.170	0.168	0.174
Total avg. abs. 3D dist. (m)	0.054	0.052	0.052
Total st.dev abs. 3D dist. (m)	0.016	0.017	0.018
Total max abs. z-axis dist. (m)	0.167	0.135	0.122
Total avg. abs. z-axis dist. (m)	0.050	0.048	0.048
Total st.dev abs. z-axis dist. (m)	0.016	0.018	0.018
Reconstructed points number	2050	8960	36,281
Time in total (sec)	1175	1233	1716

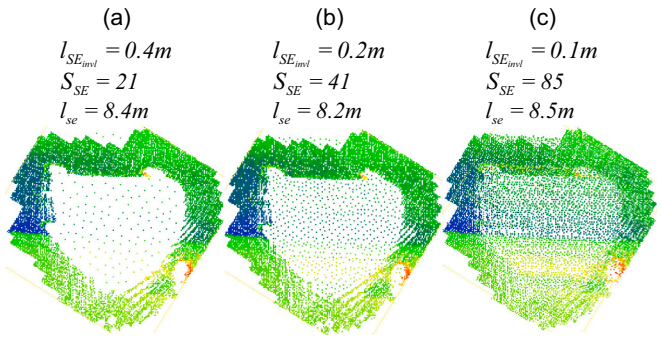


Fig. 19. Reconstructed results using MM with different combinations of l_{SEinvl} and s_{SE} (extracted and rotated clockwise from the 13th occlusion in Fig. 16).

erosion operation. In the process of generating and deleting points, the spatial query operation needs to be performed frequently, which brings a great amount of computation. The higher density can produce a clearer curb structure and slightly improve the 3D and z-axis accuracy (Fig. 19).

5. Discussion

In this section, the performance of the proposed static ground occlusion (Section 5.1) and curb detection method (Section 5.2) is analyzed based on the experimental and evaluation results, and the two occlusion reconstruction methods are compared in Section 5.3.

5.1. Performance of occlusion detection results

With the aid of α-shape based on the prior geometry knowledge of the vehicle, the boundary between the ground area and the external area of the ground is detected more accurately. In corners and boundary areas of MLS data tiles, our α -shape can better fit the true contour of the ground surface, while avoiding the exclusion of semi-close areas caused by vehicle occlusion from the ground. However, semi-close areas at the boundary of MLS data tiles may be misidentified as external areas. But this can be solved by adding adjacent MLS data tiles. The relative position relationship between vehicle and occlusion makes it easier and more efficient to locate and detect the occlusion area without traversing all ground areas. The Bridge Cell Detection algorithm successfully separated most occlusions from other adjacent occlusions, and a small number of failed cases are caused by the large Bridge between adjacent occlusions.

5.2. Performance of curb detection results

In general, the proposed curb detection method can correctly detect most of the curb areas and locally fit the real road direction well. The main problems occur in highly variable areas and at the boundaries of MLS data tiles, but the latter can be resolved by using neighbor MLS data tiles. The abnormal oriented boundary box caused by inaccurate vehicle segmentation, or the vehicle parking direction is not accurate to the road direction does not influence the detection results, which means the proposed curb detection method has good robustness to poor quality vehicle segmentation data and real street parking environments.

5.3. Comparison of two reconstruction methods

For computing speed, the cell-based linear interpolation method has a significant advantage over the point-based MM method. In terms of density and point distribution, the standard deviation of reconstruction results points out that the linear interpolation can obtain more evenly distributed reconstruction points because the generation location of candidate points can be specified by pre-defining a 2D grid, and the point density can be directly determined by the cell size. MM can only preset the linear density of SE, so it is difficult to control the overall density. As a point-based method, the distribution of reconstruction results using MM is largely affected by the distribution of points in the buffer areas. For linear interpolation, its generated points are uniformly distributed based on the interpolation grid but the point distribution in the real MLS data is not uniform, especially in areas far from the road centerline or LiDAR sensors. In addition, the predefined 2D grid allows all reconstructed points of linear interpolation to be located in the occlusion area, while MM directly processes the buffer, so some points may fall outside the occlusion area (Fig. 19). Therefore, MM reconstruction points need to be filtered in the post-processing stage. For the number of reference points required for reconstruction, linear interpolation requires at least two points (one on each side), and MM depends on the value of s_{SE}. But in general, MM needs more and more evenly distributed reference points than linear interpolation.

As can be seen from Fig. 20 (a) and (b), the reconstruction results generated by the two methods both can be well integrated into the original scanned data visually. Fig. 20 (c) to (f) show the spatial distribution of 3D and z-axis error. In general, the error distribution of linear interpolation reconstruction results is better. For the geometry of reconstructed areas (Fig. 21), the linear interpolation result is relatively smooth, because the influence of ground outliers (such as remnants of other environmental objects after segmentation) can be reduced during the interpolation operation. Compared to the overly smooth ground surface obtained by linear interpolation, the MM surface is relatively rough, but visually more consistent with the roughness of the real ground surface. However, the morphological dilation operation also processes outliers to add new outliers in the result.

5.4. Future works

After the discussion of the final occlusion detection and reconstruction performance, we find that the proposed methods still have some points that can be improved and expanded in the follow-up research. Future work will focus on the following aspects: (1) The further exploration of the MM-based reconstruction method, including the reconstruction feasibility of using other SE (such as planer SE), the speed up possibilities by using parallel or multi-core computing, the processing strategies for ground outliers, the solution for higher density reconstruction, and how to generate the points with a more uniform distribution. (2) The optimization of the road direction extraction method under complex environments, such as rugged terrain, curved roads, sparse point cloud areas, crossroads, roundabout, and other complex road sections. (3) These methods can be generalized to other types of ground occlusion, such as the occlusion caused by trees and other static environmental objects. (4) In addition to static objects, occlusion caused by dynamic objects such as moving vehicles is also of concern. (5) Finally, future work will also need to consider how to replace the 2D/3D grids in the method with point-based or TIN/TEN-based structures to avoid loss of accuracy.

6. Conclusions

This paper presents the vehicle-related ground occlusion detection and reconstruction methods for MLS data. In the occlusion detection phase, an α-shape based on prior vehicle geometry knowledge is designed to fit the actual ground boundary, and then the occlusion is rapidly detected and matched with its corresponding vehicle by using the relative position between them. In the occlusion reconstruction phase, a method of extracting curb direction from the occlusion environment is proposed to approximate the local road direction. Then, the cell-based linear interpolation and the point-based mathematical morphology respectively are used and compared to reconstruct the occlusion along the road direction.

The methodology is tested with original scanned data and evaluation data. The α-shape boundary well fits the actual ground boundary and all

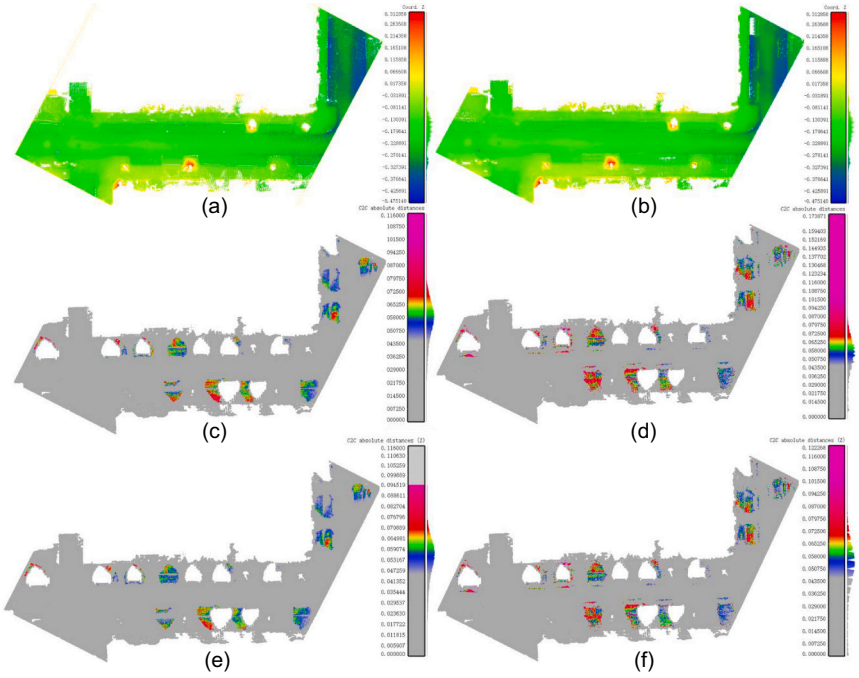


Fig. 20. Results of merging reconstructed points using linear interpolation (a) and MM (b) with the original scanned data rendered in height. 3D error of reconstructed points using linear interpolation (c) and MM (d), z-axis error of reconstructed points using linear interpolation (e) and MM (f) compared with the evaluation data (res_{grid} = 0.2 m, l_{SEinvl} = 0.1 m, and s_{SE} = 85).

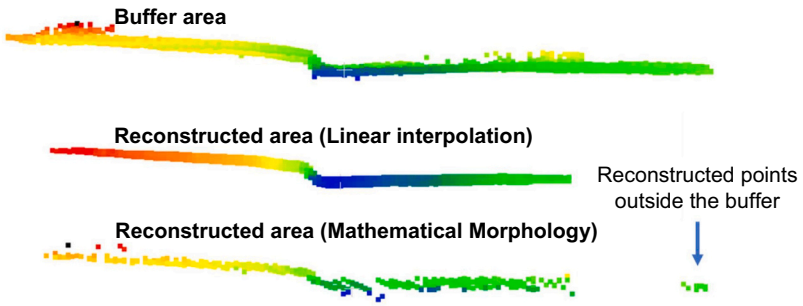


Fig. 21. The cross sections of the buffer area and reconstructed areas using MM and linear interpolation.

vehicle-related ground occlusions are correctly detected. Due to the proposed bridge detection algorithm, only one detected occlusion connects the non-ground empty area. For curb detection, the curb (road) direction is correctly extracted in most of the occluded areas. The detection method uses the relative position of the vehicle and its occlusion, but the low-quality vehicle segmentation data and inaccurate oriented boundary box of the vehicle does not affect the detection result. Two results with poor accuracy come from the boundary area of the original scanned data (can be solved by adding the next MLS data tile) and the area of dramatic height variation. The reconstruction results from both methods can visually integrate the original scanned data and recover the curb structure. Linear interpolation performs better in the error distribution. But its result is overly smoothed. The reconstruction errors of the linear interpolation method are 0.045 m in the z-axis direction and 0.051 m in total when res_{grid} is 0.2 m. The larger res_{grid} can bring a clearer curb structure and a more uniform point distribution. The roughness reconstructed by MM is closer to the real ground surface. MM also produces a more realistic distribution of points. However, the result of MM includes some issues like long calculation time. The reconstruction errors of MM are 0.048 m in the z-axis direction and 0.052 m in total when l_{SEinvl} is 0.1 m and s_{SE} is 85.

For MLS data, the proposed methods prove that the prior information about the vehicle is helpful to easily locate and detect the vehicle-related static ground occlusion and that both linear interpolation and mathematical morphology can be applied to implement the occlusion reconstruction without introducing additional data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Authors would like to thank the Xunta de Galicia given through human resources grant (ED481B-2019-061).

References

[1] S. Friedman, I. Stamos, Online facade reconstruction from dominant frequencies in structured point clouds, in: 2012 IEEE Computer Society Conference on Computer

Vision and Pattern Recognition Workshops, 2012, pp. 1–8, https://doi.org/10.1109/CVPRW.2012.6238908.
[2] J. Balado, E. González, E. Verbree, L. Díaz-Vilariño, H. Lorenzo, Automatic detection and characterization of ground occlusions in urban point clouds from mobile laser scanning data, in: ISPRS Annals of Photogrammetry Remote Sensing and Spatial Information Sciences, VI-4/W1-2020, 2020, pp. 13–20, https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-13-2020.
[3] Y. Xia, Y. Xu, C. Wang, U. Stilla, VPC-net: completion of 3D vehicles from MLS point clouds, ISPRS J. Photogramm. Remote Sens. 174 (2021) 166–181, https://doi.org/10.1016/j.isprsjprs.2021.01.027.
[4] D. Chen, R. Wang, J. Peethambaran, Topologically aware building rooftop reconstruction from airborne laser scanning point clouds, IEEE Trans. Geosci. Remote Sens. 55 (12) (2017) 7032–7052, https://doi.org/10.1109/TGRS.2017.2738439.
[5] C. Chen, B. Yang, Dynamic occlusion detection and inpainting of in situ captured terrestrial laser scanning point clouds sequence, ISPRS J. Photogramm. Remote Sens. 119 (2016) 90–107, https://doi.org/10.1016/j.isprsjprs.2016.05.007.
[6] J. Becker, C. Stewart, R.J. Radke, LiDAR inpainting from a single image, in: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, 2009, pp. 1441–1448, https://doi.org/10.1109/ICCVW.2009.5457441.
[7] A. Serna, B. Marcotegui, Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning, ISPRS J. Photogramm. Remote Sens. 93 (2014) 243–255, https://doi.org/10.1016/j.isprsjprs.2014.03.015.
[8] M. Previtali, L. Díaz-Vilariño, M. Scaioni, Indoor building reconstruction from occluded point clouds using graph-cut and ray-tracing, Appl. Sci. 8 (9) (2018) 1529, https://doi.org/10.3390/app8091529.
[9] J. Schachtschneider, A. Schlichting, C. Brenner, Assessing temporal behavior in lidar point clouds of urban environments, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 42, 2017, pp. 543–550, https://doi.org/10.15488/1694 (2017), Nr. 1W1, 42(1W1).
[10] L. Zhu, J. Hyyppa, The use of airborne and mobile laser scanning for modeling railway environments in 3D, Remote Sens. 6 (4) (2014) 3075–3100, https://doi.org/10.3390/rs6043075.
[11] Y. Gao, M. Li, Ground-based multi-platform point clouds registration, in: IOP Conference Series: Materials Science and Engineering 646, 2019, 012034, https://doi.org/10.1088/1757-899X/646/1/012034.
[12] L. Matikainen, M. Pandžić, F. Li, K. Karila, J. Hyyppä, P. Litkey, A. Kukko, M. Lehtomäki, M. Karjalainen, E. Puttonen, Toward utilizing multitemporal multispectral airborne laser scanning, Sentinel-2, and mobile laser scanning in map updating, J. Appl. Remote. Sens. 13 (4) (2019), 044504, https://doi.org/10.1117/1.JRS.13.4.044504.
[13] P. Arias, G. Facciolo, V. Caselles, G. Sapiro, A variational framework for exemplar-based image inpainting, Int. J. Comput. Vis. 93 (3) (2011) 319–347, https://doi.org/10.1007/s11263-010-0418-7.
[14] P. Buyssens, M. Daisy, D. Tschumperlé, O. Lézoray, Exemplar-based inpainting: technical review and new heuristics for better geometric reconstructions, IEEE Trans. Image Process. 24 (6) (2015) 1809–1824, https://doi.org/10.1109/TIP.2015.2411437.
[15] D. Doria, R.J. Radke, Filling large holes in LiDAR data by inpainting depth gradients, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 65–72, https://doi.org/10.1109/CVPRW.2012.6238916.
[16] W. Yuan, T. Khot, D. Held, C. Mertz, M. Hebert, PCN: Point Completion Network. 2018 International Conference on 3D Vision (3DV), 2018, pp. 728–737, https://doi.org/10.1109/3DV.2018.00088.
[17] L. Barazzetti, Point cloud occlusion recovery with shallow feedforward neural networks, Adv. Eng. Inform. 38 (2018) 605–619, https://doi.org/10.1016/j.aei.2018.09.007.
[18] T. Czerniawski, J.W. Ma, Fernanda Leite, Automated building change detection with amodal completion of point clouds, Autom. Constr. 124 (2021), 103568, https://doi.org/10.1016/j.autcon.2021.103568.
[19] J. Barros-Ribademar, J. Balado, P. Arias, S.M. González-Collazo, Visibility analysis for the occlusion detection and characterisation in street point clouds acquired with mobile laser scanning, Geocarto Int. 0 (0) (2022) 1–18, https://doi.org/10.1080/10106049.2022.2032392.
[20] J. Sun, Y. Cao, Q.A. Chen, Z.M. Mao, Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures, ArXiv:2006.16974 [Cs] (Accessed May 29, 2021), http://arxiv.org/abs/2006.16974, 2020.
[21] Y. Endo, E. Javanmardi, S. Kamijo, Analysis of occlusion effects for map-based self-localization in urban areas, Sensors 21 (15) (2021) 5196, https://doi.org/10.3390/s21155196.
[22] Y.-C. Kan, L.-T. Hsu, E. Chung, Performance evaluation on map-based NDT scan matching localization using simulated occlusion datasets, IEEE Sensors Lett. 5 (3) (2021) 1–4, https://doi.org/10.1109/LSENS.2021.3060097.
[23] B. Douillard, J. Underwood, N. Melkumyan, S. Singh, S. Vasudevan, C. Brunner, A. Quadros, Hybrid elevation maps: 3D surface models for segmentation, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 1532–1538, https://doi.org/10.1109/IROS.2010.5650541.
[24] P. Pfaff, R. Triebel, W. Burgard, An efficient extension to elevation maps for outdoor terrain mapping and loop closing, Int. J. Robot. Res. 26 (2) (2007) 217–230, https://doi.org/10.1177/0278364906075165.
[25] Y. Wang, L. Cheng, Y. Chen, Y. Wu, M. Li, Building point detection from vehicle-borne LiDAR data based on voxel group and horizontal hollow analysis, Remote Sens. 8 (5) (2016) 419, https://doi.org/10.3390/rs8050419.
[26] G.T. Toussaint, Solving geometric problems with the rotating calipers, in: Proc. IEEE MELECON'83, 83, A10, 1983. https://www.cs.swarthmore.edu/~adanner/cs97/s08/pdf/calipers.pdf (Accessed November 20, 2021).
[27] H. Edelsbrunner, D. Kirkpatrick, R. Seidel, On the shape of a set of points in the plane, IEEE Trans. Inf. Theory 29 (4) (1983) 551–559, https://doi.org/10.1109/TIT.1983.1056714.
[28] J.E. Bresenham, Algorithm for computer control of a digital plotter, IBM Syst. J. 4 (1) (1965) 25–30, https://doi.org/10.1147/sj.41.0025.
[29] G. Vosselman, L. Zhou, Detection of curbstones in airborne laser scanning data, in: Proceedings of Laser Scanning '09 : ISPRS Vol. 38 Prt 3/W8, 1–2 September 2009, Paris, France 38, 2009, pp. 111–116, in: http://www.isprs.org/proceedings/XXXVIII/3-W8/papers/p42.pdf (Accessed November 18, 2021).
[30] G. Matheron, J. Serra, The birth of mathematical morphology, in: Proceedings of 6th International Symposium on Mathematical Morphology, Sydney, Australia, 2002, 2002, pp. 1–16. https://smil.cmm.minesparis.psl.eu/publis/C-72.pdf (Accessed March 29, 2022).
[31] J. Balado, P. van Oosterom, L. Díaz-Vilariño, M. Meijers, Mathematical morphology directly applied to point cloud data, ISPRS J. Photogramm. Remote Sens. 168 (2020) 208–220, https://doi.org/10.1016/j.isprsjprs.2020.08.011.