PLOS | ONE

RESEARCH ARTICLE

A real-time road detection method based on reorganized lidar data

Fenglei Xu, Longtao Chen, Jing Lou, Mingwu Ren*

School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

* mingwuren@163.com

Abstract

Road Detection is a basic task in automated driving field, in which 3D lidar data is commonly used recently. In this paper, we propose to rearrange 3D lidar data into a new organized form to construct direct spatial relationship among point cloud, and put forward new features for real-time road detection tasks. Our model works based on two prerequisites: (1) Road regions are always flatter than non-road regions. (2) Light travels in straight lines in a uniform medium. Based on prerequisite 1, we put forward difference-between-lines feature, while *ScanID* density and obstacle radial map are generated based on prerequisite 2. According to our method, we construct an array of structures to store and reorganize 3D input firstly. Then, two novel features, difference-between-lines and *ScanID* density, are extracted, based on which we construct a consistency map and an obstacle map in Bird Eye View (BEV). Finally, the road region is extracted by fusing these two maps and refinement is used to polish up our outcome. We have carried out experiments on the public KITTI-Road benchmark, achieving one of the best performances among the lidar-based road detection methods. To further prove the efficiency of our method on unstructured road, the visual outcomes in rural areas are also proposed.

OPEN ACCESS

Citation: Xu F, Chen L, Lou J, Ren M (2019) A real-time road detection method based on reorganized lidar data. PLoS ONE 14(4): e0215159. https://doi.org/10.1371/journal.pone.0215159

Editor: Yanyong Guo, University of British Columbia, CANADA

Received: October 29, 2018

Accepted: March 18, 2019

Published: April 16, 2019

Data Availability Statement: The data underlying the results presented in the study are available from (https://github.com/xflxfl1992/RDR).

Funding: This work was supported by the National Major Scientific Research Instrument Development Project (Grant Number: 61727802). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Introduction

Traversable road detection is always a core task in the context of autonomous driving vehicles, which has been studied for decades. Bunches of implementations are proposed based on various sensors, in which vision-based road detection is the most conventional kind. However, the lack of depth information makes environmental perception inadequate to construct a perfect road model. So, a consensus has been reached that range data is necessary in road detection, which leads to researches on utilization of 3D information recently. Range data is usually provided by stereo-cameras, radars or lidars. In off-road environment, which is much more challenging compared to well-structured urban scenes, 3D lidar scanners are more widely needed.

3D point cloud processing based on lidar boom recently, due to the rapid development of lidar. The resolution of lidar starts to increase from centimeter level to millimeter level, while solid-state lidar is also being on mass production. 3D lidar scanners can capture geometry in a very high resolution, while not suffering from external illumination, which is a big trouble to image-based road detection. Although the high resolution of 3D lidar data makes it possible to detect various obstacles and traversable areas for unmanned vehicles, it brings mass calculation to searching or indexing tasks. Therefore, it is meaningful to design a model to transform the discrete and unorganized lidar point cloud into an organized form and cut computation reasonably. So, we involve a more organized lidar form with abundant information in this paper.

Traditional lidar-based road detection is constructed in view of flat road plane assumption. However, taking unstructured road into account, we believe that this assumption is fragmentary. According to our view, road region is just relatively flatter than non-road regions. Gentle ups and downs are frequently existed on road, while mild obstacles are also permitted especially in rural road scenes. This is the base, on which we put forward difference-between-lines feature and use adaptive threshold to locate road boundary point. Another prerequisite concerned is that light travels in straight lines in a uniform medium. So laser would be blocked by obstacles, which causes a high point density before a positive obstacle with an undetected region generated after it. Obstacles and the undetected regions behind pose a threat to safe autonomous driving, so we would like to describe both of them as obstacles in our work.

In this paper, we put forward a new organization mode of lidar data, which generates spatial relationship among points. *ScanID* and *PointID* are defined to directly describe this relationship, which are quantification of scanner pitch angle and lidar rotation angle to an extent. Next, two new features, difference-between-lines and *ScanID* density, are proposed based on this organized lidar data presentation, which are robust to structured as well as unstructured road detection. Finally, novel obstacle and road detection algorithms are generated by utilizing these two features respectively. Our method enjoys low cost of computing, meeting real-time requirement in the field of autonomous driving.

The rest of this paper is organized as follows. The normal existing methods and our approach are introduced in Section Materials and Methods. Section Experiments shows the experiments and corresponding results of our approach, as well as the comparison with other existing methods. We draw our conclusion in Section Conclusion. Note that this paper is best viewed in color.

Materials and methods

Existing road detection methods

Road detection, a core task in intelligent transportation research ([1] [2]), has been studied for decades. Various models and methods based on different sensors are conducted to deal with this task in all kinds of scenes. [3] presents a survey of recent progress in road detection research. All models can be mainly divided into 2 trends:

1. approaches based on single sensor (monocular camera, stereo vision or lidar).
2. approaches based on combining multiple senors. ([4] [5])

It is reasonable to combine different inputs and construct a complete map of the environment. However, it is beyond the scope of this paper, in which we only talk about single input.

Among the single-camera based road detection methods ([6] [7] [8]), most of them are targeted to design classifiers, in which color ([9] [10]) and texture information ([11] [12]) are usually applied as main features. Lane marks ([13] [14]) are also frequently used on structured road detection, which are distinct especially on newly built asphalt road. However, these methods are sensitive to illumination or road shape. Meanwhile, flat plane assumption failure has significant impact on these models, especially in off-road scenes. The stereo-vision methods ([15] [16]) solves parts of these problems by employing the disparity map acquired by stereo matching. The U-V-disparity maps [17] and local descriptors [18] generated from disparity map can be used to analyze road regions [19]. Different from single-camera, stereo-vision can estimate distance through disparity maps. However, disparity error increases with distance, while dense stereo matching [20], needed by these models, are time consuming.

Compared to single-camera and stereo-vision, 3D lidar sensors outperform at measuring distance. Various methods are put forward on point cloud processing to deal with multifarious tasks, especially on object detecting and classifying ([21] [22]). Most of these models squash point cloud down to a 2D image-like format, because numerous processing methods based on image can be used after decades of researches on image processing. Recently, direct 3D data operating methods ([23] [24]) are studied by some scholars, but they are tentative and too time-consuming, which is not suitable to road detection task. Flat plane assumption ([25] [26]) is widely applied in 3D lidar based models. Flat areas are usually considered as road regions in high-definition lidar data. Road boundaries ([27] [28]) are also used frequently to locate road regions. Besides, a widely used road detection approach is proposed in [29] by estimating the min-max evaluation map. [30] presents a survey of grid map based road and obstacle detection methods for unmanned ground vehicles. Besides to grid map, projecting lidar points onto image plane ([31] [32]) is another expression of lidar data. In approach [33], lidar-histogram is proposed, which helps detect road in lidar-imagery. These well-known methods are mainly based on height difference between road and non-road regions. If the height difference is not salient, such as in off-road scenes, clearer data organization and stabler features are expected.

Approach

To overcome the feedbacks of existing lidar based methods, as well as satisfying real-time requirements, we rearrange point cloud data, based on which a road consistency map and an obstacle map are generated. Road consistency map takes advantage of spatial relationship among point cloud, while obstacle map makes full use of height difference between road and non-road regions. Combining these two confidence maps, road region could be well segmented both in structured urban and unstructured rural scenes.

A. Rearrangement of lidar data. In order to deal with road detection task, we rearrange lidar data at first step to accomplish a presentation mode with spatial relationship and low complexity. Here, we use Velodyne HDL-64 S2 lidar, which is a core 3D lidar sensor of autonomous driving vehicles. It contains 64 laser heads, which are separated into 2 blocks with 32 lasers respectively. All laser heads are settled on a spinning base, providing a 360° horizontal field of view and 26.8° vertical field of view. The original raw output data of lidar are arranged on spherical coordinate system, mainly including rotation angle θ, measurement distance d, and radiation intensity I. Calibration file provided by manufacturer contains the pitch angle ϕ of each laser head. When we convert the (d, θ, ϕ) in spherical coordinate system to the (x, y, z) in Cartesian coordinates, approximately 1.33 million sampling points are detected per second, so it is quite necessary to rearrange the 3D lidar data before processing.

We construct an array of structures to store lidar data, in which each structure contains all information of one sampling point, including its coordinates (x, y, z), intensity I, which scan line it belongs to (*ScanID*) and the point ranking of such point (*PointID*) in this scan line. Here, we convert a point in spherical coordinate to Cartesian coordinates by

$$\begin{cases} x = d\cos\phi\cos\theta \\ y = d\cos\phi\sin\theta \\ z = d\sin\phi \end{cases} \tag{1}$$

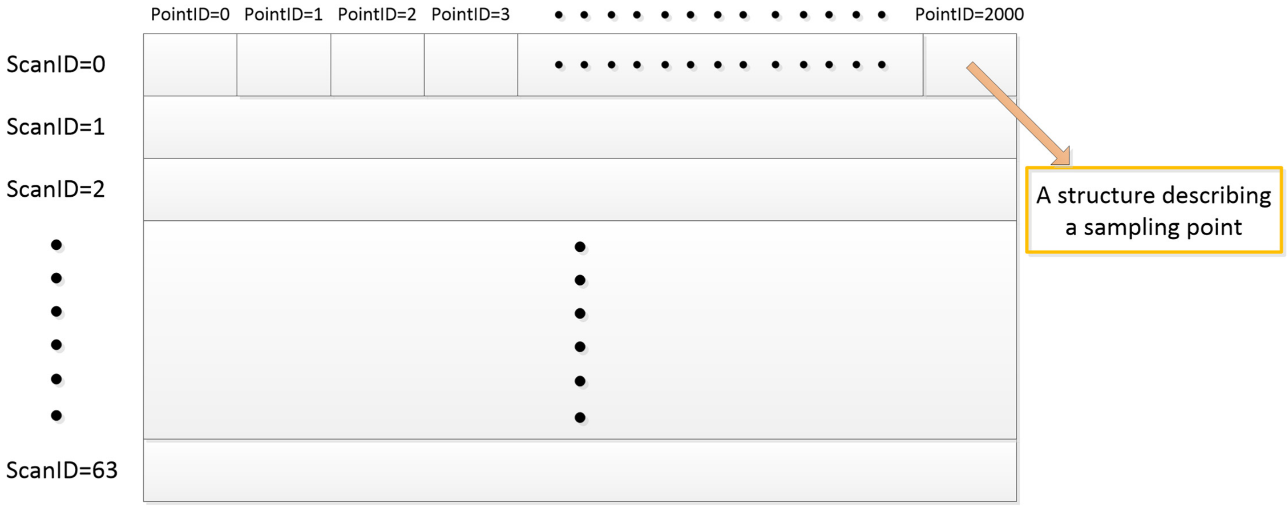


Fig 1. Array of structures.

https://doi.org/10.1371/journal.pone.0215159.g001

ScanID indicates which laser head does the sampling point belong to. It ranks from 0 to 63 based on the pitch angle ϕ of the corresponding laser head. When lidar is settled on a big flat plane, No.0 scan line indicates the nearest one from lidar, while No.63 is the farthest one. *PointID* is concerned to the rotation angle θ of this point. Here, we define one lap of lidar data as a frame, and the rotating speed of lidar is settled at 10*Hz*. Then, a frame of lidar data contains 64 scan lines with approximately 2000 sampling points in each line, because the angular resolution of Velodyne HDL-64 S2 is 0.18° at 10*Hz*. Moreover, we use *ScanID* and *PointID* to arrange our array of structure. Our lidar data array is constructed as Fig 1. Each structure in this array represents a point, including 7 different features, *x*, *y*, *z*, *d*, *I*, *ScanID* and *PointID*, where *x*, *y*, *z* indicate the point location in Cartesian coordinate system, while *d* represents the linear distance, and the last 2 features describe adjacent relationship among points. Here, in our method, we mainly use the spatial information from lidar, so intensity *I* is kept but not used.

B. Consistency map. Road can be divided into 2 types simply, structured and unstructured roads. Normally, the structured roads have clear borders or strong difference with non-road regions, while the unstructured ones share weak natural edges with non-road regions. So, it is not quite reasonable to divide road regions simply based on boundary in all instances. In that case, we put forward a new pattern, distance-between-lines, to describe road region, which takes road undulating into account by using neighbor relationship among point cloud. No matter structured roads or unstructured ones, neighbor relationship is stable in both cases. However, unlike image presentation, it is quite difficult to discover neighbor relationship between scan lines by original 3D lidar data. Here, we witness an advantage of our rearrangement of lidar data, because sampling points in the same column have similar rotation angle, constructing spatial relationship between lines.

During our unstructured road experiments, we found that although road region become undulating in country road scenes, it shows a similar trend between scan lines along the road, which also occurs on structured roads (Fig 2). In Fig 2, three adjacent scans from structured road and three from unstructured are shown in two columns. It is clear to observe that 3 adjacent scans are similar, no matter in structured roads or unstructured ones. We call this consistency of a road. Then, we compute the distance between scans to smooth the road region, while amplify the chaos in non-road regions. The black scan in Fig 2 indicates the Δd

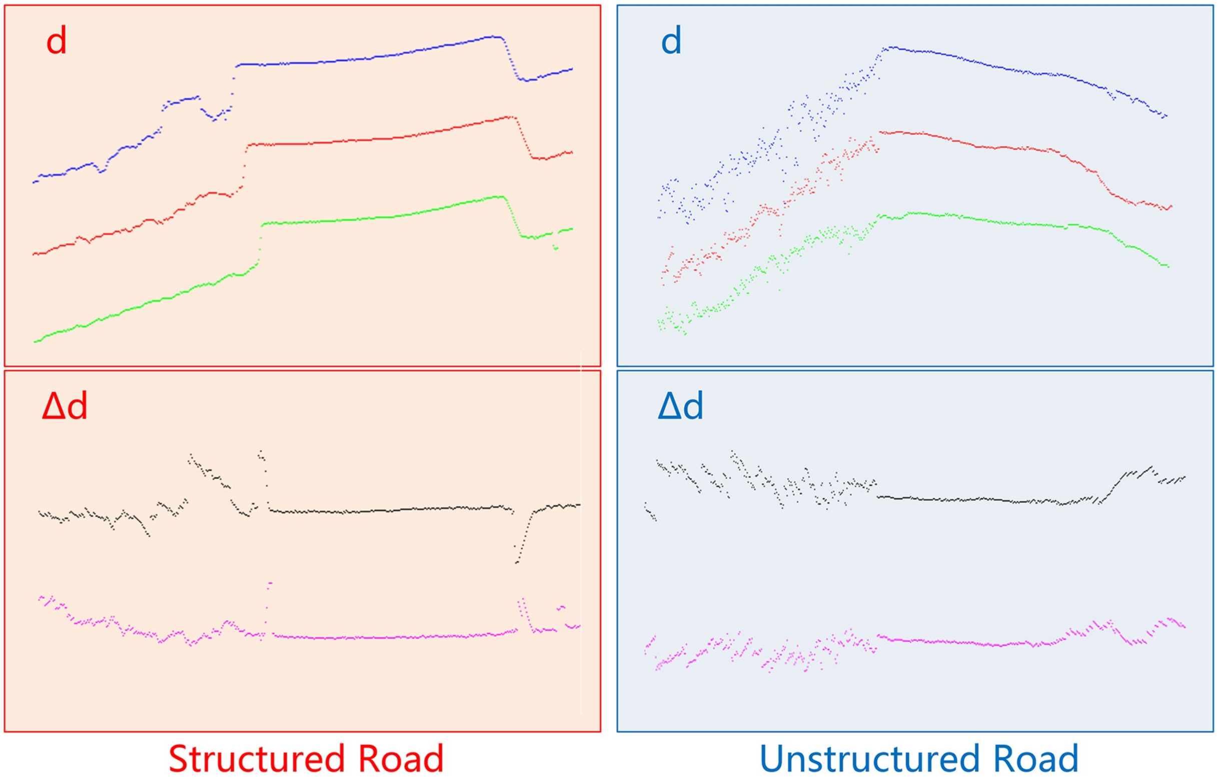


Fig 2. *d* and Δ*d* on structured and unstructured roads.

https://doi.org/10.1371/journal.pone.0215159.g002

computed by the blue one minus the red one, while the purple scan comes from subtracting the green scan from the red one. Here, we witness that a much clearer road region could be segmented from each scan both in structured and unstructured road scenes. So, based on our structure array, we put forward a distance-between-lines feature to describe road consistency. Distance-between-lines (Δ*d*) is defined as follows.

$$\Delta d_i = |d_{ij} - d_{ik}| \tag{2}$$

Here, parameter *i* indicates *PointID*, while *j* and *k* represent *ScanID*, which are next to each other. *d* represents the distance from lidar center to such point. Using Δd_i, we reduce the impact of ups and downs on unstructured road, while remain or even enhance the messy situation on non-road regions. We suppose that road is always flatter than non-road regions in a scene, which means the variance of Δ*d* on road region is relatively smaller than those on non-road regions. This is because *d* values are in chaos on grass, rocks and so on, which lead to Δd_i also in a jumble on these areas. As the *PointID* is arranged by rotation angle θ, structured edges will be amplified by Δ*d* rater than suppressed, unless lidar is right over the edge. We plot 3 neighbor scan lines from structured and unstructured roads on Fig 2, in which we can clearly see that, after computing Δ*d* between scans, the road region is much clearer to extract in both types of road. Moreover, the non road part witness more chaos, due to superposition of disorder.

In order to find candidate road regions, we separate each scan line into parts with suitable length by finding the inflection points. We use a low computational complexity way to locate the inflection points on each scan by the formula below:

$$\left|\Delta d_i - \sum_{j=i-k}^{i+k} \mathcal{W}_j \times \Delta d_j\right| > thre \tag{3}$$

where

$$\mathcal{W}_j = \frac{\exp(-\frac{(j-i)^2}{2\sigma^2})}{\sum_{j=i-k}^{i+k} \exp(-\frac{(j-i)^2}{2\sigma^2})} \tag{4}$$

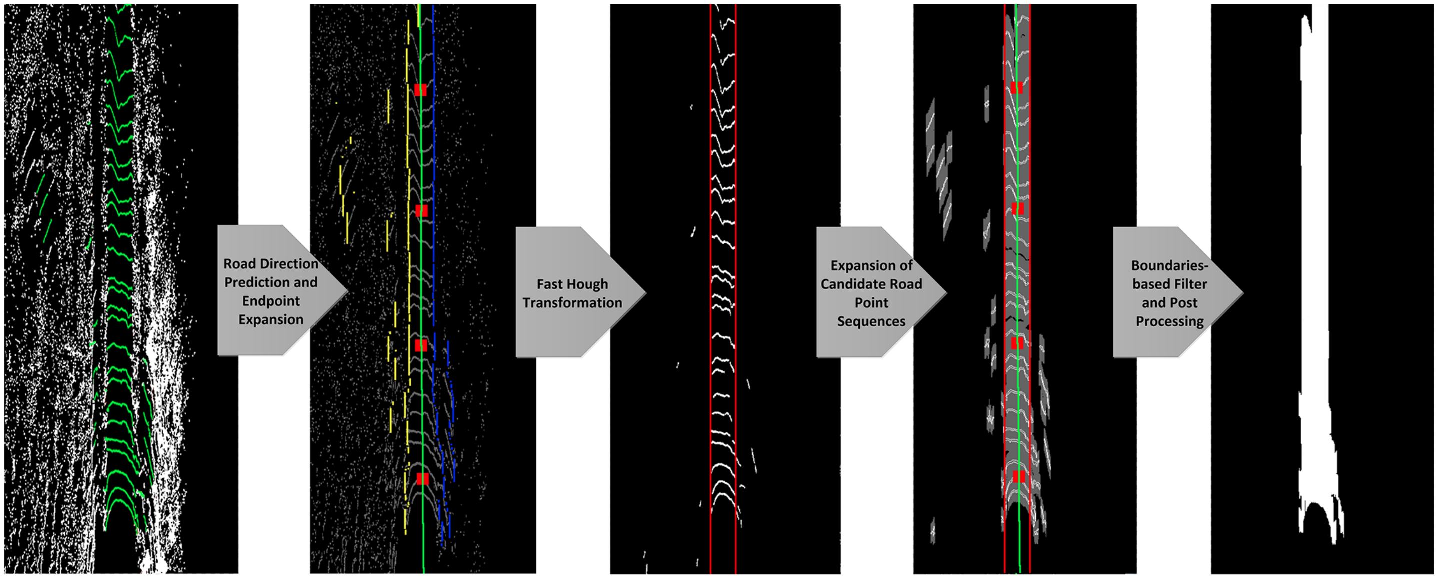
i and *j* indicate *PointID*, and *k* is filter radius. Here, we use the Gaussian filtering consequence as an adaptive interface, and inflection points are defined as those whose distance from the interface exceed threshold. To simplify operation, Gaussian template is used. Next, we use minimum-variance to evaluate the suitable point sequences on each scan line. Loss function is defined as:

$$\mathcal{L} = \frac{1}{n-m} \sum_{i=m}^{n} \|\Delta d_i - \overline{\Delta d}\| \tag{5}$$

m, *n* indicate the inflection points located before, which determine a point sequence on a scan line. $\overline{\Delta d}$ is the average value of Δd_i. Here, in order to find all suitable *m*, *n* on each line, we could limit the road width by transcendental knowledge. Moreover, the value of $\mathcal{L}$ is also limited by a empirical threshold to select candidate road parts from each scan line.

After locating the candidate point sequences on each scan, we project point cloud to a grid map shown in Fig 3a (candidate point sequences are colored in green), and the endpoints of such sequence are supposed to be candidate road boundary points in the grid map. Due to sparsity of lidar scans, we expand the endpoints on each scan along forward direction, then the end points of road regions become vertical line segments (Fig 3b). However, when we encounter to a line segment, it is not clear that it belongs to a left boundary or a right one, because lidar scan lines are circles in the grid map. Therefore, we predict the road direction by finding the midpoint of the longest continuous suitable point sequence on each scan in front of lidar. Moreover, in order to reduce computation, we choose 4 scans, instead of computing all 64 scans to predict the road direction. Just as the green line in Fig 3b, it is the road direction we predict, and the red squares are the 4 midpoints we choose. By road direction, we can easily decide the left boundary points (shown in yellow in Fig 3b) and the right ones (shown in blue in Fig 3b). Then, fast Hough transformation is adopted to determine the best lines, which go through most left or right boundary points, as the left and right boundary respectively (Fig 3c). Next, we expand all candidate road point sequences on each scan along forward direction as shown in gray in Fig 3d. The two road boundaries and the road direction could help to locate the true road regions from the candidate ones. Here, we believe that all gray pixels between boundaries are true road area, and the gray pixels linked to true road area but outside the detected boundaries are also believed to be road region. To polish up the outcome, we fill in the small blanks among gray pixels. Due to the sparsity of point cloud, we expend the farthest road region (with largest *ScanID*) along the nearest boundary direction to borders of the grid map, where is no laser echo. Finally, we acquire an image of road region by consistency measurement, which is called consistency map $\mathcal{M}_{Consistency}$ here (Fig 3e). In case of transformation from BEV to perspective, we store the corresponding height of each pixel in $\mathcal{M}_{Consistency}$, coming from our array of structures.

C. Obstacle map. Based on the novel array of structures mentioned in subsection A, a grid map is constructed by projecting 3D points to BEV, uniquely using *ScanID* density as the


Road Direction Prediction and Endpoint Expansion
Fast Hough Transformation
Expansion of Candidate Road Point Sequences
Boundaries-based Filter and Post Processing

（a） Candidate Road Region Point Sequences
（b） Candidate Road Boundary Segments
（C） Believed Road Boundaries
（d） Tentative Road Regions
（e） Consistency Map

Fig 3. Consistency map $\mathcal{M}_{Consistency}$ generating flow.

https://doi.org/10.1371/journal.pone.0215159.g003

value of each grid. This coordinate system is image-like and gray-level of a pixel (x, y) represents the number of different *ScanID* that this grid contains. Dimensionality reduction from 3D to 2D and resolution reduction by generating grids hugely decrease computation afterwards, which is extremely significant to autonomous driving.

Here, instead of measuring the height of each grid, which is sensible to shaking and parameter calibration, we use another pattern to describe obstacles. According to our experiments, we discover a phenomenon that different types of obstacles on road causes different effects on scan lines, as can be seen in Fig 4. When scanning a road plane, the measured distances on one scan share nearly the same value. However, when points encounter a positive obstacle, the measured distances of these points are shorter than those on road plane. Oppositely, the distances from points on negative obstacles to laser head are longer. Water hazards will cause specular reflection, refraction and absorption on laser pulse, which leads to an undetected region without signal received by the lidar receiver. Dangerous water hazards on road always occur in a form of negative obstacle, so we mainly target on detecting positive and negative obstacles in this paper.

According to Fig 4, lidar scan lines witness distortions when crossing positive and negative obstacles. The distance between two scan lines becomes much more narrow at obstacles. Furthermore, several lines could even gather at the back edge of negative obstacles and the front of positive ones. Here, we put forward a new pattern called *ScanID* density, which indicates the number of different *ScanID* in one grid. We use this pattern to generate an obstacle grid map to describe positive and negative obstacles at the same time. As the grid map in Fig 5 shown, pixels in green represent laser points in this grid are from the same scan line, while red pixels indicate that there are laser points from different scan lines in one grid. So, obstacles are easily located on pixel level in the grid map, in which red pixels are obstacles and green ones indicate relatively flat plane, while black regions are undetected areas.

As we all known, rays propagate along a straight line, so there are always undetected regions (shown in black in Fig 5) behind the obstacles (shown in red). The true road region in a perspective is the final target of our road detection model, and pixels of an obstacle in camera

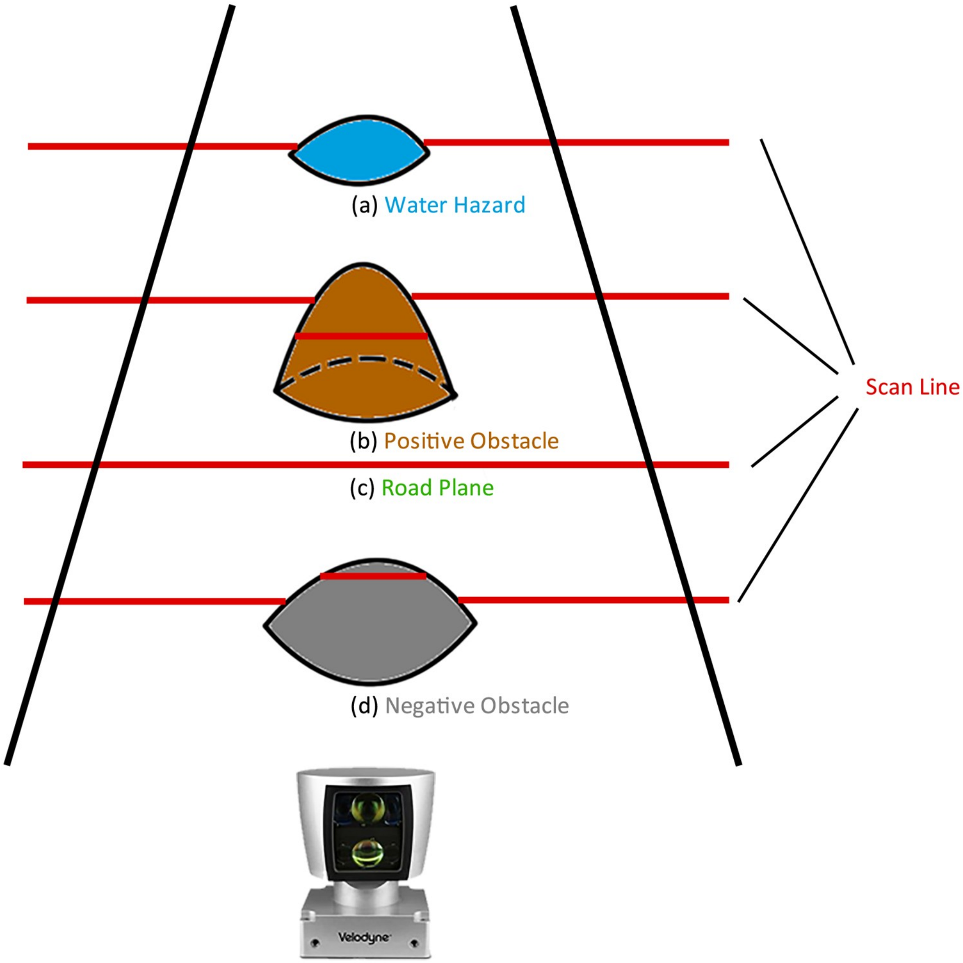


Fig 4. Different types of obstacles detected on road.

https://doi.org/10.1371/journal.pone.0215159.g004

view including the red pixels and the black region behind in the corresponding grid map. So we introduce an obstacle radial map to describe obstacles in the grid map, which is also in conformity with BEV form. The obstacle radial map is generated by Algorithm 1.

Algorithm 1 generating obstacle radial map $\mathcal{M}_{Radial}$

```
Input: grid map $\mathcal{M}_{Grid}$.
Output: obstacle radial map $\mathcal{M}_{Radial}$.
for each $\theta \in (-90°, 90°)$ do
  $Flag \leftarrow false$;
  for each $x$ from lidar center $(x_0, y_0)$ to $\mathcal{M}_{Grid}$ boundaries do
    $y = \tan\theta \times (x - x_0) + y_0$;
    if $\mathcal{M}_{Grid}(x, y) = red$ then
      $Flag \leftarrow true$; $Continue$;
    else if $\mathcal{M}_{Grid}(x, y) = green$ then
      $Flag \leftarrow false$; $Continue$;
    end if
    if $Flag$ then
      $\mathcal{M}_{Grid}(x, y) = gray$;
    end if
  end for
end for
```

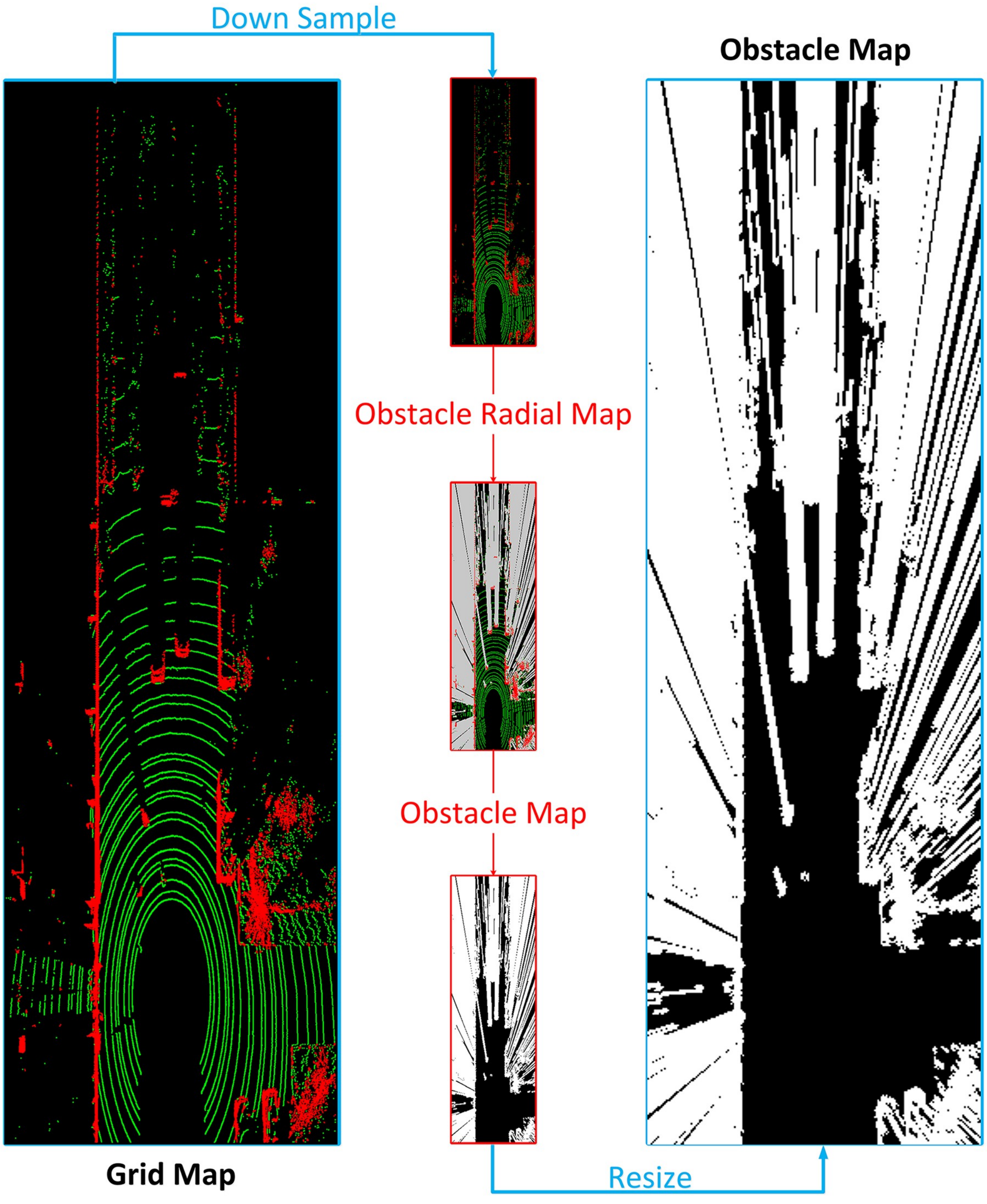


Fig 5. Obstacle map generating procedure.

https://doi.org/10.1371/journal.pone.0215159.g005

To sum up, we search all directions along radial rays from lidar center, and classify obstacle pixels and undetected pixels behind to obstacle regions. In order to accelerate obstacle radial map generating and filter the input simultaneously, we down sample the input grid map by 16 times. Next, obstacle regions, the red and gray pixels, are extracted from the obstacle radial map. Then, after a resizing procedure, the obstacle map $\mathcal{M}_{Obstacle}$ is finally obtained. The obstacle map generating procedure is shown in Fig 5.

D. Combination and refinement.

Algorithm 2 refinement for the fused map $\mathcal{M}_{Fused}$

```
Input: $\mathcal{M}_{Fused}$.
Output: refined fused map $\mathcal{M}_{Fused}$.
  1. $\mathcal{M}_{Fused} = \mathrm{PROJECTION}(\mathcal{M}_{\mathrm{Fused}})$ → project to perspective
  2. $\mathcal{M}_{Fused} = \mathrm{ROI}(\mathcal{M}_{\mathrm{Fused}})$
  3. $M_{Fused} = \mathrm{ADJUST}(\mathcal{M}_{\mathrm{Fused}})$ → spatial morphological filtering
  4. Keep the regions which reach the bottom, left or right border of the map $\mathcal{M}_{Fused}$
  5. $\mathcal{M}_{Fused} = \mathrm{HOLE}-\mathrm{FILL}(\mathcal{M}_{\mathrm{Fused}})$
```

In order to obtain the ultimate road region pixels, based on which we can decide accessible area in autonomous driving tasks, two maps determined by distance-between-lines feature and obstacle detection respectively are fused to one, which makes full use of both road plane information and obstacle information simultaneously. The fused binary map is named as $\mathcal{M}_{Fused}$, which is defined as follows.

$$\mathcal{M}_{Fused} = \mathcal{M}_{Consistency} \cap \overline{\mathcal{M}}_{Obstacle} \tag{6}$$

Next, we project $\mathcal{M}_{Fused}$ to perspective for further refinement and evaluation. The details of refinement procedure is shown in Algorithm 2. Besides, the whole processing flow is shown in Fig 6.

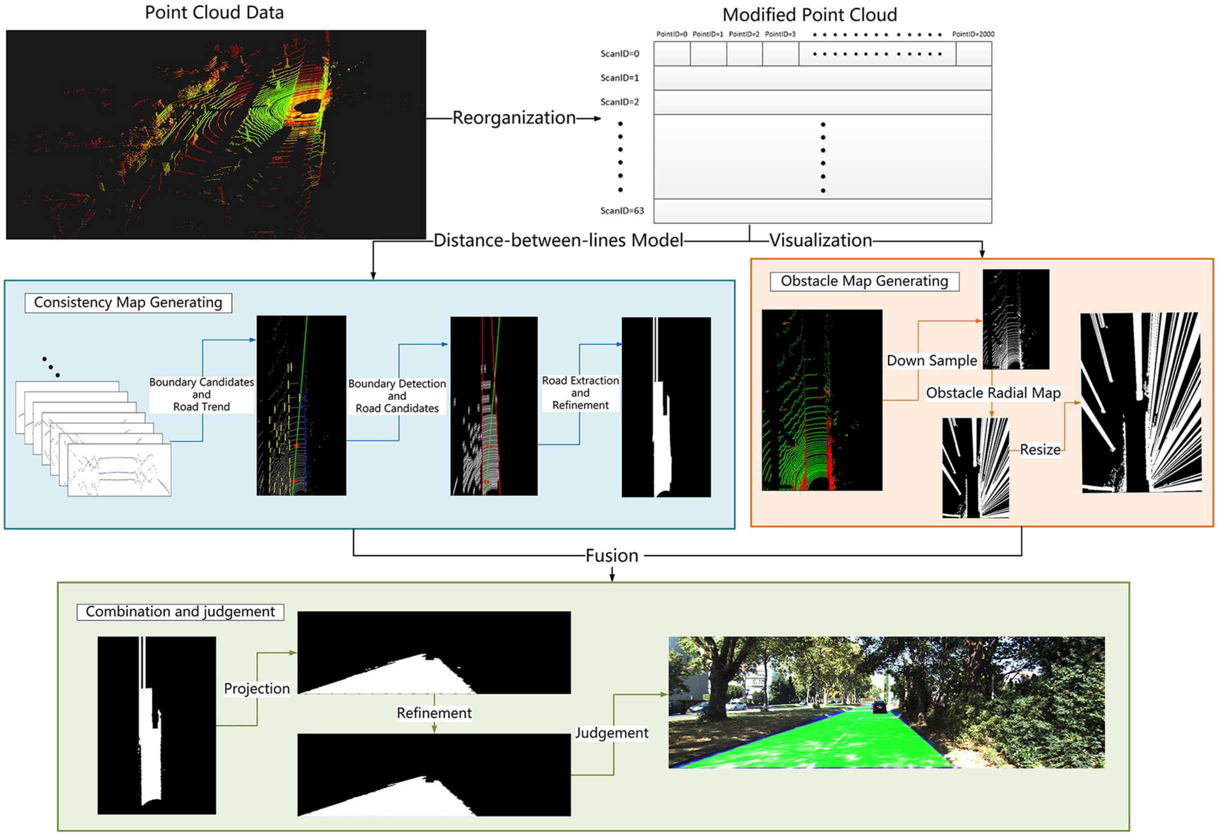


Fig 6. Processing flow.

https://doi.org/10.1371/journal.pone.0215159.g006

Results

In this section, we evaluate the proposed method with 5 well-performed models demonstrated on KITTI dataset [34], including Road Estimation with Sparse 3D Points From Velodyne (RES3D-Velo) [31], Graph Based Road Estimation using Sparse 3D Points from Velodyne (GRES3D+VELO) [35], CRF based Road Detection with Multi-Sensor Fusion (FusedCRF) [36], LidarHistogram (LidarHisto) [33], Hybrid Conditional Random Field (HybridCRF) [37]. To easily express our method, we name it RDR, which indicates a Road Detection Method based on Reorganized Lidar Data.

A. Dataset

In order to evaluate our proposed method, experiments are conducted on a PC by C/C++ with a CPU (2.5GHz) and a 8GB size memory without any code optimization, operated on a public dataset (KITTI). KITTI road dataset contains 289 marked and 290 unmarked road scenes for method evaluation.

We use the same evaluation criteria as KITTI, including MaxF (Maximum F1-measure), AP (Average Precision), PRE (Precision), REC (Recall), FPR (False Positive Rate), FNR (False Negative Rate), to compare our method with other models. The definition of MaxF, PRE,

Table 1. Performance comparison with other methods on KITTI.

Method	Benchmark	MaxF	AP	PRE	REC	FPR	FNR
RES3D-Velo	UM_ROAD	83.81%	73.95%	78.56%	89.80%	11.16%	10.20%
	UMM_ROAD	90.60%	85.38%	85.96%	95.78%	17.20%	4.22%
	UU_ROAD	83.63%	72.58%	77.38%	90.97%	8.67%	9.03%
	URBAN_ROAD	86.58%	78.34%	82.63%	90.92%	10.53%	9.08%
GRES3D+VELO	UM_ROAD	85.43%	83.04%	82.69%	88.37%	8.43%	11.63%
	UMM_ROAD	88.19%	88.65%	83.98%	92.85%	19.48%	7.15%
	UU_ROAD	84.14%	80.20%	80.57%	88.03%	6.92%	11.97%
	URBAN_ROAD	86.07%	84.34%	82.16%	90.38%	10.81%	9.62%
FusedCRF	UM_ROAD	89.55%	80.00%	84.87%	94.78%	7.70%	5.22%
	UMM_ROAD	89.51%	83.53%	86.64%	92.58%	15.69%	7.42%
	UU_ROAD	84.49%	72.35%	77.13%	93.40%	9.02%	6.60%
	URBAN_ROAD	88.25%	79.24%	83.62%	93.44%	10.08%	6.56%
lidarHisto	UM_ROAD	89.87%	83.03%	91.28%	88.49%	3.85%	11.51%
	UMM_ROAD	93.32%	93.19%	95.39%	91.34%	4.85%	8.66%
	UU_ROAD	86.55%	81.13%	90.71%	82.75%	2.76%	17.25%
	URBAN_ROAD	90.67%	84.79%	93.06%	88.41%	3.63%	11.59%
HybridCRF	UM_ROAD	90.99%	85.26%	90.65%	91.33%	4.29%	8.67%
	UMM_ROAD	91.95%	86.44%	94.01%	89.98%	6.30%	10.02%
	UU_ROAD	88.53%	80.79%	86.41%	90.76%	4.65%	9.24%
	URBAN_ROAD	90.81%	86.01%	91.05%	90.57%	4.90%	9.43%
RDR (ours)	UM_ROAD	92.61%	86.14%	91.62%	93.62%	3.90%	6.38%
	UMM_ROAD	93.37%	89.97%	93.73%	93.02%	6.84%	6.98%
	UU_ROAD	89.64%	81.98%	87.72%	91.65%	4.18%	8.35%
	URBAN_ROAD	92.22%	86.50%	91.59%	92.87%	4.70%	7.13%

Our method has a novel outcome compared with models demonstrated on KITTI dataset.

https://doi.org/10.1371/journal.pone.0215159.t001

REC, FPR and FNR are as follows:

$$PRE = \frac{|M \cap G|}{|M|}, REC = \frac{|M \cap G|}{|G|}, \tag{7}$$

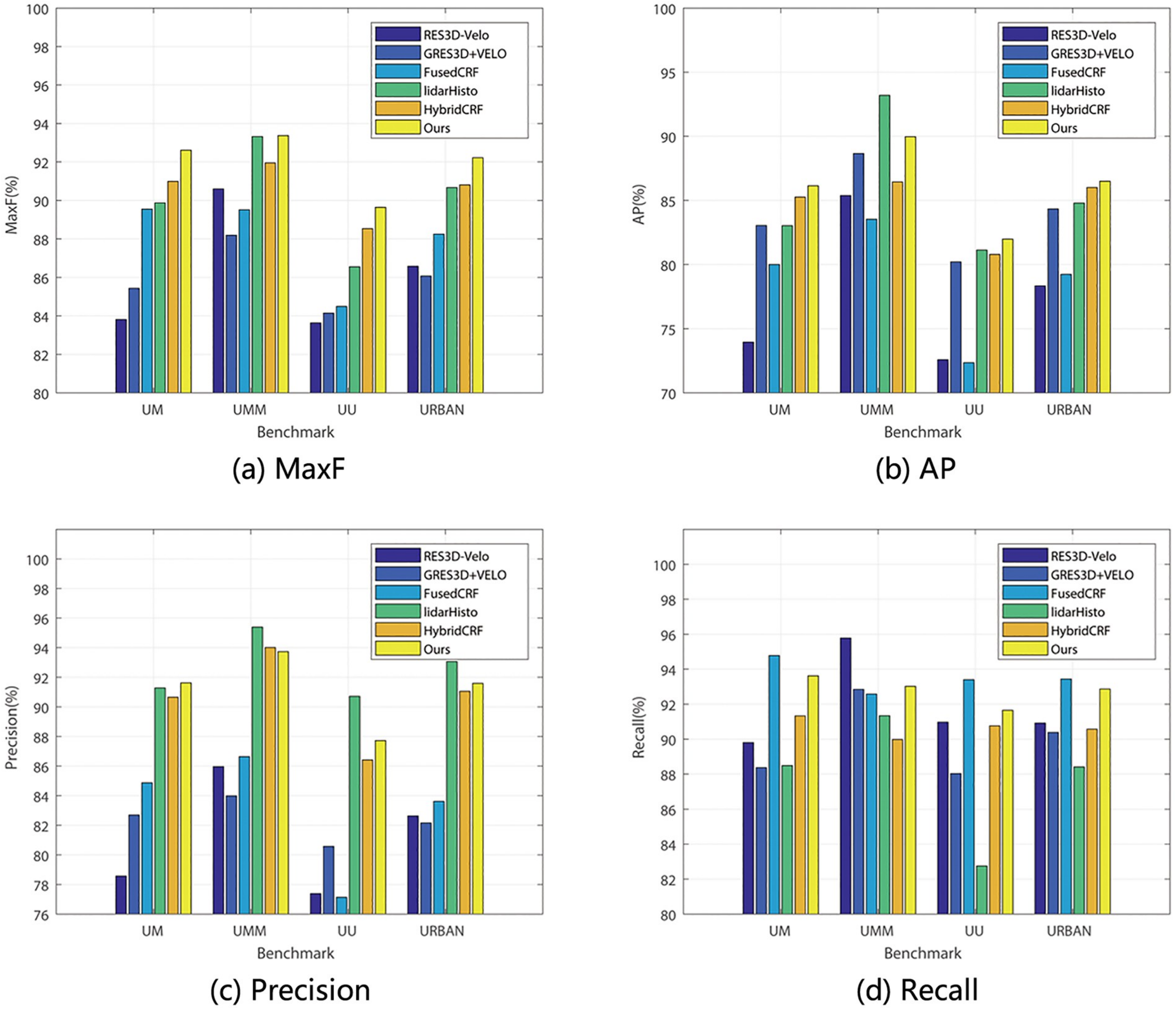
$$MaxF = \max\left(\frac{2 \times PRE \times REC}{PRE + REC}\right) \tag{8}$$

$$FPR = \frac{number\ of\ false\ positives}{number\ of\ negative\ instances}, FNR = \frac{number\ of\ false\ negatives}{number\ of\ positive\ instances} \tag{9}$$

where M indicates a binary segmentation, while G represents the ground truth of such instance.

B.Comparison

In order to prove the efficiency of our method, we compare it with other road detection models on KITTI dataset, which utilize lidar information, including RES3D-Velo [31],



(a) MaxF

(b) AP

(c) Precision

(d) Recall

Fig 7. Bar charts of MaxF, AP, Precision and Recall comparison.

https://doi.org/10.1371/journal.pone.0215159.g007

Table 2. Time efficiency comparison.

Method	Runtime	Environment
RES3D-Velo	360ms	1 core @ 2.5 Ghz (C/C++)
GRES3D+VELO	60ms	4 cores @ 2.8 Ghz (C/C++)
FusedCRF	2000ms	1 core @ 2.5 Ghz (C/C++)
lidarHisto	100ms	2 core @ 2.5 Ghz (C/C++)
HybridCRF	1500ms	1 core @ 2.5 Ghz (C/C++)
RDR (ours)	60ms	2 core @ 2.5 Ghz (C/C++)

Our method outperforms other models demonstrated on KITTI dataset on time efficiency.

https://doi.org/10.1371/journal.pone.0215159.t002

GRES3D+VELO [35], FusedCRF [36], LidarHisto [33], HybridCRF [37], to evaluate the validity of our method. A performance comparison is shown in Table 1 and bar charts of significant indexes (Fig 7) are also proposed for clearer observation. According to Table 1, our method generates much more accurate outcomes than other models on UM_Road, UU_ROAD and URBAN_ROAD benchmarks. On UMM_ROAD benchmark, LidarHisto performs well and almost catch up with RDR, lagging behind 0.05% only.

The time efficiency comparison has been shown in Table 2, in which our method performs the first place among the methods listed. In general, our model obtains a state-of-art result on KITTI dataset with low time cost. The typical visual results of these models are also listed in Fig 8. To further evaluate the efficiency of RDR on unstructured road, experiments are taken on the data acquired by ourselves. Real vehicle tests are carried out, and 30 typical rural road scenes are chosen to form a small dataset with road region tags. Our model achieves 84.47%, 81.79%, and 87.33% on MaxF, PRE and REC respectively on this dataset. The typical visual results of rural road scenes are proposed in Fig 9.

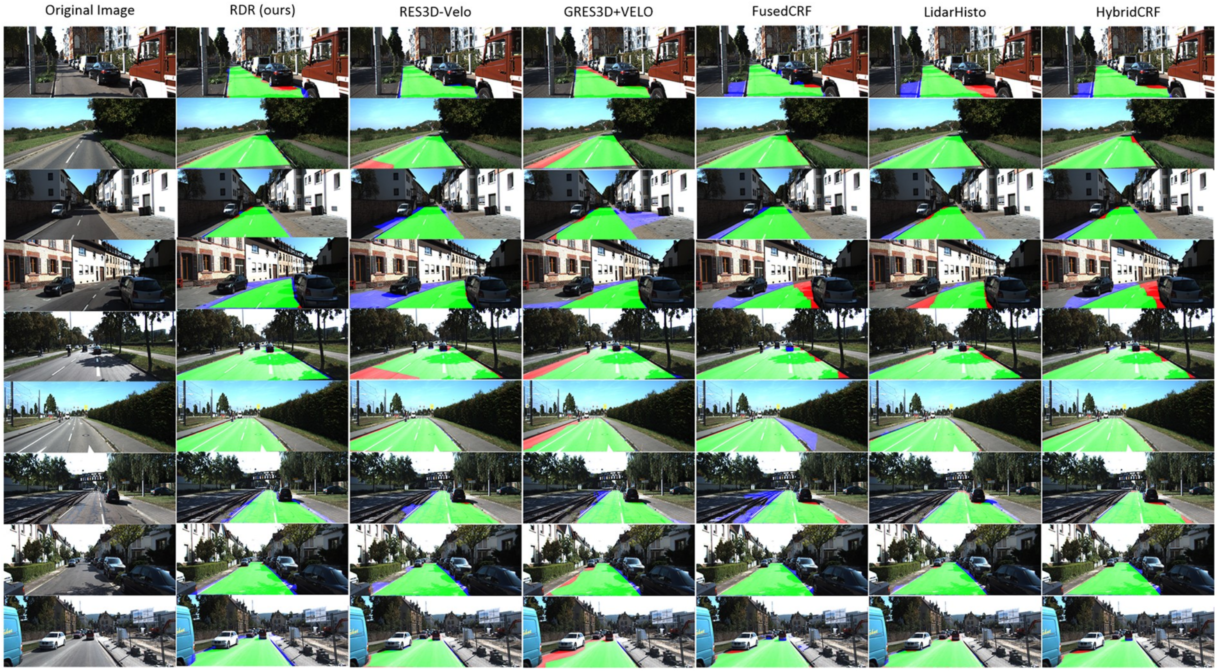


Fig 8. Visual results comparison.

https://doi.org/10.1371/journal.pone.0215159.g008

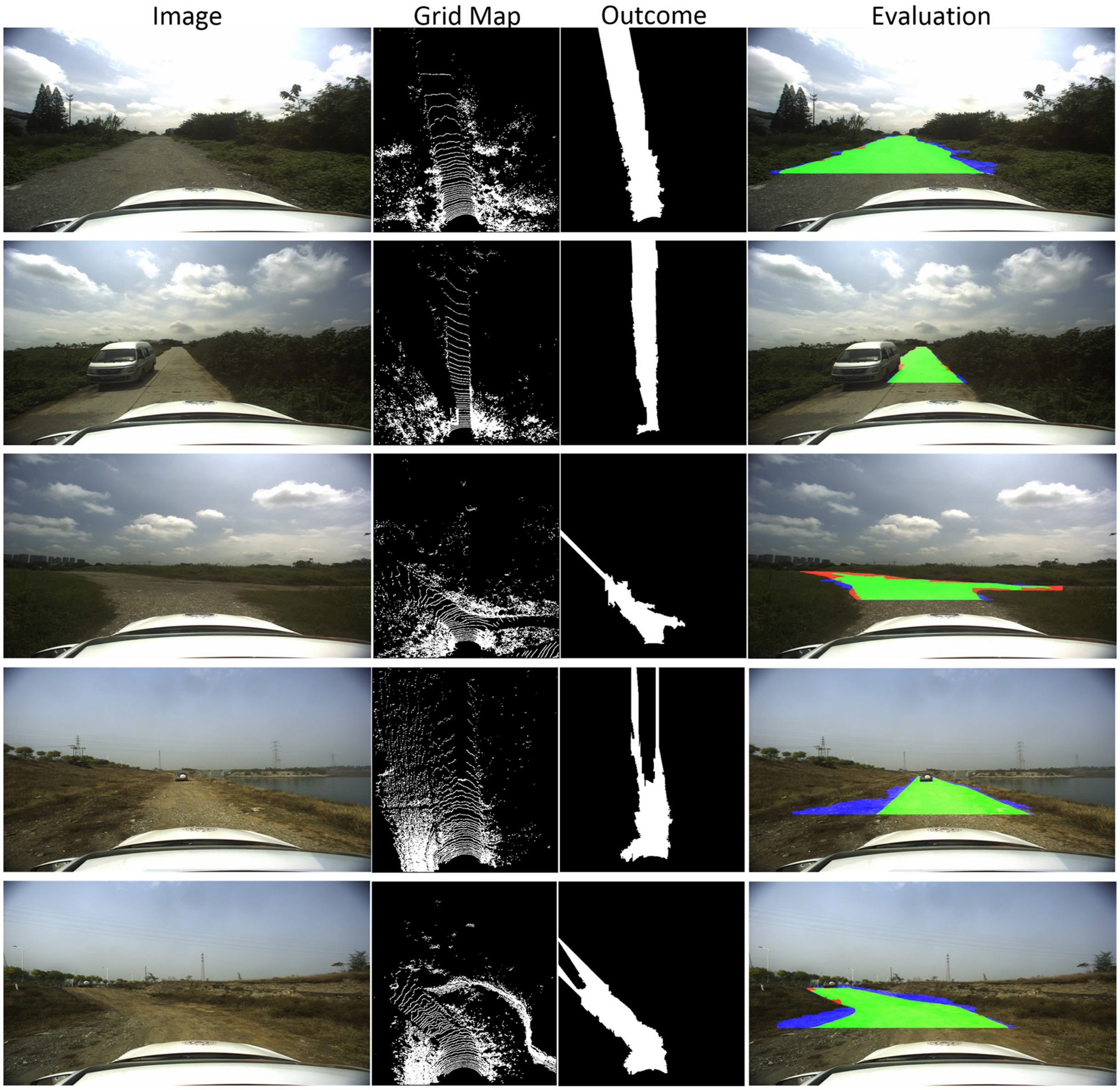


Fig 9. Visual results of RDR on unstructured road.

https://doi.org/10.1371/journal.pone.0215159.g009

Discussion

Although RDR performs well and exceeds the compared models on datasets, it does fail in some cases. These failures are mainly caused by two attributes of our method. Fig 10 shows the typical failure cases collected from KITTI dataset. Fig 10(a) is mainly caused by failure of assumption 1. The non-road region is a flat plane, and there is no inflection point at road boundaries, so that RDR select a point sequence containing both road and non-road regions as one candidate road part. Fig 10(b) occurs because we use straight line fitting to locate road boundaries, which are used for choosing road region from candidates in RDR. To reduce this influence, we increase the vertical resolution in grid map to stretch road along forward direction and reduce bend radian. Fig 10(b) is the outcome after adopting this step.

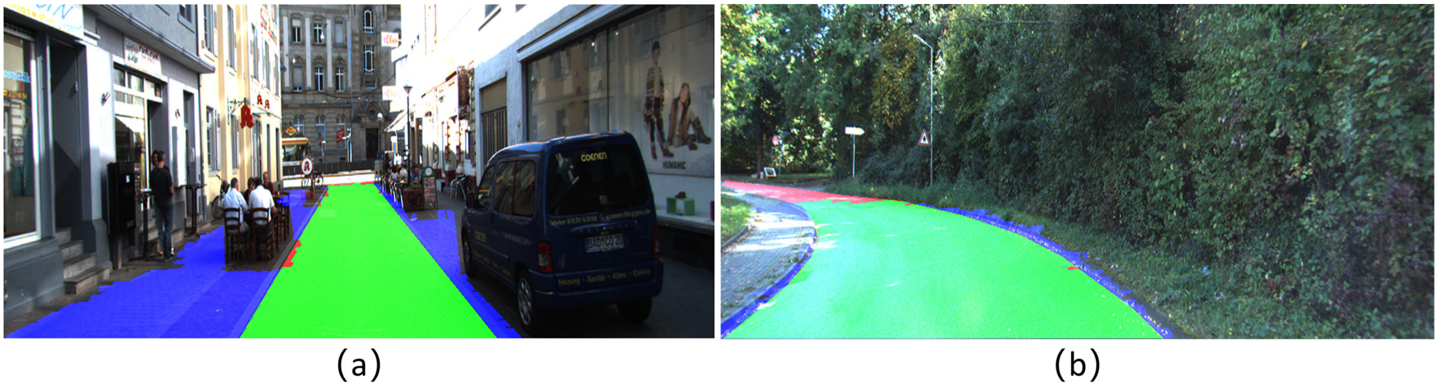


Fig 10. Fail instances.

https://doi.org/10.1371/journal.pone.0215159.g010

Conclusion

Throughout this paper, we present a novel road detection method based on modified lidar data, which meets real-time requirement. The reorganization of lidar data constructs spatial relationship based on *ScanID* and *PointID*, which are actually quantifications of laser head pitch angle and lidar rotation angle to some extent. Based on the reorganized lidar data, two main innovations of our work, distance-between-lines and *ScanID* density, are put forward to generate consistency map and obstacle map respectively. Next, two maps are fused to one, and after refinement, road region is detected ultimately. Experimental results indicate a novel performance of our proposed model.

With regard to future work, multi-sensor fusion is under consideration, which can make environment modeling much more perfect. For example, failure in Fig 10(a) could be solved, if camera input is combined. Furthermore, a certain degree of information redundancy ensures the safety of autonomous vehicle. Second, we intend to use lidar information more deeply. Dimensionality reduction and ignorance of intensity abandon a lot of precious information from point cloud. So that, employing intensity and making use of point cloud directly would be our focus of attention afterwards.

Acknowledgments

This work was supported by the National Major Scientific Research Instrument Development Project (Grant Number: 61727802).

Author Contributions

Conceptualization: Fenglei Xu.

Data curation: Fenglei Xu.

Formal analysis: Fenglei Xu.

Methodology: Longtao Chen.

Project administration: Mingwu Ren.

Resources: Jing Lou.

Software: Fenglei Xu, Longtao Chen, Jing Lou.

Supervision: Mingwu Ren.

Validation: Fenglei Xu.

Visualization: Fenglei Xu.

Writing – original draft: Fenglei Xu.

Writing – review & editing: Fenglei Xu, Longtao Chen, Jing Lou.

References

1. Ma C, Hao W, Wang A, Zhao H. Developing a Coordinated Signal Control System for Urban Ring Road Under the Vehicle-Infrastructure Connected Environment. IEEE Access. 2018 09; 6:1–1.
2. Changxi M, Ruichun H, Wei Z, Xiaolei M. Path optimization of taxi carpooling. PLOS ONE. 2018 08.
3. Hillel AB, Lerner R, Dan L, Raz G. Recent progress in road and lane detection: a survey. Machine Vision & Applications. 2014; 25(3):727–745. https://doi.org/10.1007/s00138-011-0404-2
4. Caltagirone L, Bellone M, Svensson L, Wahde M. LIDAR-Camera Fusion for Road Detection Using Fully Convolutional Neural Networks. CoRR. 2018;abs/1809.07941.
5. Hu X, Rodriguez FSA, Gepperth A. A multi-modal system for road detection and segmentation. In: Intelligent Vehicles Symposium Proceedings; 2014. p. 1365–1370.
6. Tan C, Hong T, Chang T, Shneier M. Color model-based real-time learning for road following. In: Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE; 2006. p. 939–944.
7. Alvarez JM, Gevers T, Lecun Y, Lopez AM. Road Scene Segmentation from a Single Image. Springer Berlin Heidelberg; 2012.
8. Xu F, Hu B, Chen L, Wang H, Xia Q, Sehdev P, et al. An illumination robust road detection method based on color names and geometric information. Cognitive Systems Research. 2018; 52:240–250. https://doi.org/10.1016/j.cogsys.2018.06.019
9. He Y, Wang H, Zhang B. Color-based road detection in urban traffic scenes. IEEE Transactions on Intelligent Transportation Systems. 2004; 5(4):309–318. https://doi.org/10.1109/TITS.2004.838221
10. Wang B, Fremont V, Rodriguez SA. Color-based road detection and its evaluation on the KITTI road benchmark. In: Intelligent Vehicles Symposium Proceedings; 2014. p. 31–36.
11. Kong H, Audibert JY, Ponce J. General Road Detection From a Single Image. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society. 2010; 19(8):2211. https://doi.org/10.1109/TIP.2010.2045715 PMID: 20371404
12. Kong H, Audibert JY, Ponce J. Vanishing point detection for road detection. Computer Vision and Pattern Recognition. 2013;p. 96–103.
13. Li H, Nashashibi F. Robust real-time lane detection based on lane mark segment features and general a priori knowledge. In: IEEE International Conference on Robotics and Biomimetics; 2011. p. 812–817.
14. Wu PC, Chang CY, Lin CH. Lane-mark extraction for automobiles under complex conditions. Pattern Recognition. 2014; 47(8):2756–2767. https://doi.org/10.1016/j.patcog.2014.02.004
15. Liu M, Shan C, Zhang H, Xia Q. Stereo Vision Based Road Free Space Detection. In: International Symposium on Computational Intelligence and Design; 2017. p. 272–276.
16. Zhang Y, Su Y, Yang J, Ponce J, Kong H. When Dijkstra Meets Vanishing Point: A Stereo Vision Approach for Road Detection. IEEE Transactions on Image Processing. 2018; PP(99):1–1.
17. Hu Z, Uchimura K. U-V-disparity: An efficient algorithm for stereovision based scene analysis; 2005.
18. Wang K, qu L, Chen L, Gu Y, Zhang X. Non-flat Road Detection Based on A Local Descriptor. arXiv preprint. 2016 09;arXiv:1609.08436.
19. Labayrade R, Aubert D, Tarel JP. Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In: Intelligent Vehicle Symposium; 2002. p. 646–651 vol.2.
20. Geiger A, Roser M, Urtasun R. Efficient large-scale stereo matching. In: Asian Conference on Computer Vision; 2010. p. 25–38.
21. Korah T, Medasani S, Owechko Y. Strip Histogram Grid for efficient LIDAR segmentation from urban environments. In: Computer Vision & Pattern Recognition Workshops; 2011.
22. Meng X, Currit N, Wang L, Yang X. Detect Residential Buildings from Lidar and Aerial Photographs through Object-Oriented Land-Use Classification. Photogrammetric Engineering & Remote Sensing. 2012; 78(1):35–44. https://doi.org/10.14358/PERS.78.1.35
23. Charles RQ, Hao S, Mo K, Guibas LJ. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: IEEE Conference on Computer Vision Pattern Recognition; 2017.

24. Su H, Jampani V, Sun D, Maji S, Kalogerakis V, Yang MH, et al. SPLATNet: Sparse Lattice Networks for Point Cloud Processing. 2018 02.

25. Moosmann F, Stiller C. Joint self-localization and tracking of generic objects in 3D range data. In: IEEE International Conference on Robotics and Automation; 2013. p. 1146–1152.

26. Gu S, Lu T, Zhang Y, Alvarez JM, Yang J, Kong H. 3-D LiDAR + Monocular Camera: An Inverse-Depth-Induced Fusion Framework for Urban Road Detection. IEEE Transactions on Intelligent Vehicles. 2018; 3(3):351–360. https://doi.org/10.1109/TIV.2018.2843170

27. Fardi B, Scheunert U, Cramer H, Wanielik G. Multi-modal detection and parameter-based tracking of road borders with a laser scanner. In: Intelligent Vehicles Symposium, 2003. Proceedings. IEEE; 2003. p. 95–99.

28. Wijesoma WS, Kodagoda KRS, Balasuriya AP. Road-boundary detection and tracking using ladar sensing. IEEE Transactions on Robotics & Automation. 2004; 20(3):456–464. https://doi.org/10.1109/TRA.2004.825269

29. Thrun S, Montemerlo M, Dahlkamp H, Stavens D, Aron A, Diebel J, et al. Stanley: The Robot That Won the DARPA Grand Challenge; 2006.

30. Papadakis P. Terrain traversability analysis methods for unmanned ground vehicles: A survey. Engineering Applications of Artificial Intelligence. 2013; 26(4):1373–1385. https://doi.org/10.1016/j.engappai.2013.01.006

31. Shinzato PY, Wolf DF, Stiller C. Road Terrain Detection: Avoiding Common Obstacle Detection Assumptions Using Sensor Fusion. In: Intelligent Vehicles Symposium (IV); 2014.

32. Gu S, Zhang Y, Yang J, Kong H. Lidar-based urban road detection by histograms of normalized inverse depths and line scanning. In: European Conference on Mobile Robots; 2017. p. 1–6.

33. Chen L, Yang J, Kong H. Lidar-histogram for fast road and obstacle detection. In: 2017 IEEE International Conference on Robotics and Automation (ICRA); 2017. p. 1343–1348.

34. Fritsch J, Kuehnl T, Geiger A. A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms. In: International Conference on Intelligent Transportation Systems (ITSC); 2013.

35. Shinzato PY. Estimation of obstacles and road area with sparse 3D points. Institute of Mathematics and Computer Science (ICMC) / University of Sao Paulo (USP); 2015.

36. Xiao L, Dai B, Liu D, Hu T, Wu T. CRF based Road Detection with Multi-Sensor Fusion. In: Intelligent Vehicles Symposium (IV); 2015.

37. Xiao L, Wang R, Dai B, Fang Y, Liu D, Wu T. Hybrid conditional random field based camera-LIDAR fusion for road detection. Information Sciences. 2018; 432:543–558. https://doi.org/10.1016/j.ins.2017.04.048