DARTBOARD BASED GROUND DETECTION ON 3D POINT CLOUD

Leonardo Gigli, Beatriz Marcotegui, Santiago Velasco-Forero

MINES ParisTech, Université PSL, Centre de Morphologie Mathématique (CMM), 77300 Fontainebleau, France

Commission II, WG 3

KEY WORDS: Point cloud processing, ground detection, bird eye view, adaptive grid, quasi-flat zones.

ABSTRACT:

3D laser scanners acquire 3D point clouds of real environments. The process consists in sampling the scene with laser beams rotating around an axis. By construction, the point density decreases with the distance to the scanner. This density heterogeneity is a major issue, in particular for mobile systems in the context of autonomous driving, as usually a single scan is processed simultaneously (instead of mapping applications that can integrate several scans, reducing the density heterogeneity). We propose a dartboard grid with cell size increasing radially in order to adapt the grid size to the point density. The effectiveness of this strategy is demonstrated by means of a ground detection task, a fundamental step in many workflows of analysis of 3D point clouds.

1. INTRODUCTION

Ground detection is a fundamental problem to solve for several applications such as 3D modeling process and mobile robot navigation. In autonomous-driving applications, the interest for ground detection is twofold. The first is to narrow down the zone of the scene where the vehicle can navigate through. The second is the fact that once removed the ground from the set, other objects can be identified as isolated components. Indeed, this strategy is employed in many object detection or object classification algorithms.

Some approaches project the 3D point cloud into 2D images, highly reducing the computational complexity. However, the uniform grid usually used for this purpose is not adapted to the acquisition configuration because the point density decreases with the distance to the scanner. In this paper we propose a *dartboard* grid that fits the sampling scheme of the scanner. The aim is to avoid the over-segmentation of far away objects if the cell size is too small or the loss of details of close objects if the cell size is too large.

This paper is organised as follows. Section 2 reviews previous works on ground detection tasks. Section 3 introduces our proposal, including a *dartboard* representation that simulates the acquisition system configuration, resulting in an ideal representation for further processing. Then, section 4 demonstrates the effectiveness of our method and compares it to several state of the art methods like RANSAC, λ-flat zones, and CNN-based methods on Semantic KITTI dataset. Finally, section 5 concludes this work and provides some perspectives.

2. PREVIOUS WORKS

Many 3D object detection and classification approaches start with a ground detection step (Serna and Marcotegui, 2013, Serna and Marcotegui, 2014, Roynard et al., 2016). The idea is motivated by the fact that once removed the ground from the scene all the other objects in the scene appear as different connected components. The pipeline is illustrated in Figure 1.

Among the approaches proposed in the literature for ground detection, many solutions rely on geometrical intuitions. A simple attempt to solve this problem is to model the ground as a flat surface and carry out a planar approximation using RANSAC paradigm introduced by (Fischler and Bolles, 1981). Examples of RANSAC based approaches are (Oniga et al., 2007, Schnabel et al., 2007, Gallo et al., 2011). In spite of the fact that those methods are robust to outliers, the assumption of the ground as a unique plane is not realistic, even in the urban context. To solve this problem, (Hernández and Marcotegui, 2009, Serna and Marcotegui, 2013) proposed to use λ-flat zones to detect the ground in dense point clouds. The method projects the point cloud on a regular grid parallel to the xy plane placed at the lowest value of z coordinate, and stores for each grid cell the value of the minimal elevation among all projected points on the same pixel. This is called the *Bird's Eye View* (BEV). Once obtained the projected BEV images the segmentation via λ-flat zones is carried out to obtain the ground. Similarly, (Roynard et al., 2016) project points on a discrete horizontal grid and the z value with the highest value in the histogram is selected as ground seed. Then a region growing approach is used to detect the ground. Both methods are very similar: lambda flat zone and region growing approaches rely on the same hypothesis of smooth height variation. The unique difference is the initialization step.

These methods were proposed for a mobile mapping application with a relative density homogeneity. They are sensitive to the grid resolution and the authors suggest carefully picking a value that allows both to obtain one point per pixel in the average case and to obtain connected profiles in the projected image. Unfortunately this assumption does not hold for standard autonomous driving applications. In that case, the scanner is mounted on top of the car and the axis of the scanner is orthogonal to the ground. The resulting point density decreases with the distance from the scanner. In this kind of scenario it is not possible to find a good resolution value. On one hand, a sufficiently big resolution disconnects objects in the projected image. On the other hand a small resolution accumulates too many points in pixels closer to the scanner where the point cloud density is high and the corresponding information aggregation may disturb the detection of small objects.

A more recent method has been introduced by (Zhang et al., 2016). Their idea is to turn upside down the point cloud and let

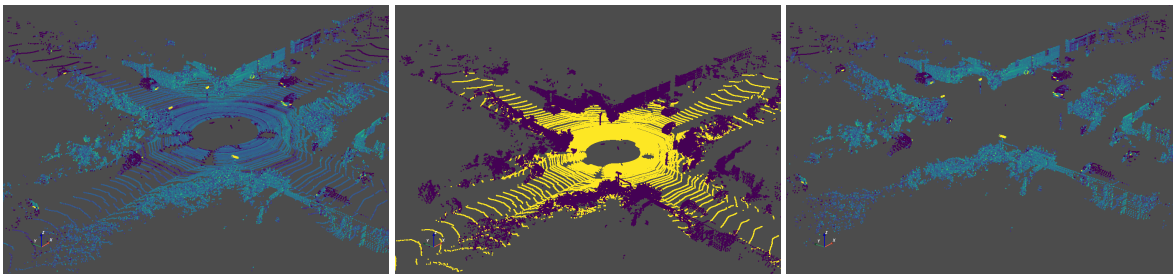


Figure 1. Common classification pipeline: first ground is detected. Then, object classification is simplified.

drop a cloth to the inverted surface from above. The ground is then detected analysing the intersections between the nodes of the cloth and the inverted point cloud. Finally, in recent years several CNN-based methods have been introduced in the more general problems of semantic segmentation of a 3D point cloud (Landrieu and Simonovsky, 2018, Thomas et al., 2019, Hu et al., 2020). Concerning the ground detection task (Velas et al., 2018) propose to project the point cloud using a spherical view and generate 2D images containing range, z and laser intensity values. The resulting images are then used to train a fully CNN (FCNN) in order to obtain a binary segmentation. Finally the labels are back projected to 3D points. This kind of approach has been also used by (Behley et al., 2019) and (Milioto et al., 2019) to carry out a semantic segmentation of the scene.

Polarnet (Zhang et al., 2020) is an interesting recent work that introduces an improved BEV image representation. The proposed grid contains two axes: radius and azimuth angle, assuming the matrix is connected on both ends of the radius axis. Polarnet demonstrates a more homogeneous distribution of points in the new grid representation compared to the Cartesian grid and achieves improved results compared to the state of the art. However they do not consider different ring thicknesses, accounting for the higher sparsity in distant areas. Moreover the larger size of distant sectors (due to longer azimuthal circular sector perimeter) is not taken into account in their polar representation. A similar idea is proposed in (Zhu et al., 2021), where a cylindrical partition, with a polar pattern in the horizontal plane, is proposed. As in (Zhang et al., 2020), the ring thickness is uniform in the radial axis.

In this paper we go further in the BEV grid definition. We propose a *dartboard* shaped grid that better adapts to the scanner configuration. The resulting improved BEV representation is a better starting point for any analysis approach. We demonstrate the effectiveness of the method on a simple ground detection scheme that does not require an annotated database or a learning procedure. The resulting approach outperforms other classical techniques.

3. GROUND DETECTION ON POINT CLOUDS WITH HETEROGENEOUS DENSITY

Assuming that the height variations in ground area are smooth, (Hernández and Marcotegui, 2009) proposes to detect the ground as the largest quasi-flat (Meyer, 2001, Soille, 2008) of the BEV range image.

A quasi-flat zone, also named λ-flat zone (λ-FZ), is defined follows:

Definition 1. **λ-flat zone:** Two neighboring pixels p, q belong to the same λ-flat zone of a function f , if their difference $|f_p - f_q|$ is smaller than or equal to a given λ value

$$\forall (p,q) \text{ neighbors} : |f_p - f_q| \leq \lambda$$

Let us now present our proposed method. It uses λ-flat zones to detect ground on Point Clouds and aims to solve the problems deriving from the high variation of point density in the scene. By construction, the point density of 3D Point Clouds decreases with the distance to the scanner. This means that projecting the points on a squared grid defined over the xy plane, the pixels far from the scanner have a higher probability to be empty. This causes a problem of connection between peripheral pixels. Figure 2 illustrates this problem. Left column shows BEV of ground pixels of a SemanticKITTI frame and right column the corresponding ground detection based on the largest quasi-flat zone ((Hernández and Marcotegui, 2009)). The resolution of the images in the first row is 1 m side. 15% of 3D ground points are missing in this detection. The problem worsens with higher resolutions. The resolution of the images in the second row is 20 cm side, where 22% of 3D ground points are missing.

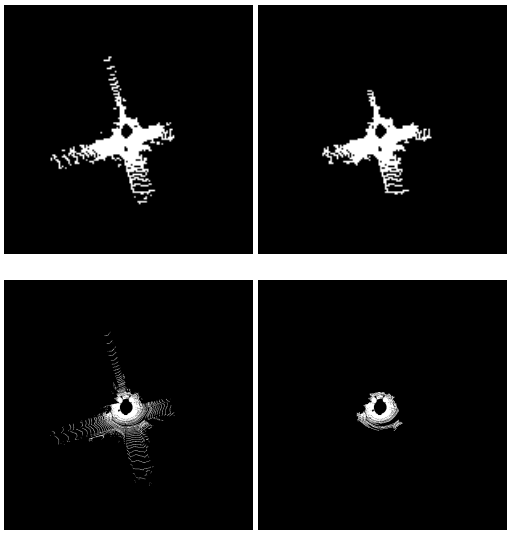


Figure 2. Ground disconnection caused by peripheral lower density. Ground detected (right) by the method described in (Hernández and Marcotegui, 2009) and its corresponding ground truth (left). Two different resolutions: pixels of 1 m side (first row) and 20 cm side (second row).

Our method consists in splitting the xy grid as a particular polar grid that we will introduce in section 3.2. In order to adapt the cell grid to the point density, its size increases with the distance to the scanner. Then, BEV image is interpolated in each circular sector. The method uses I_{min}, I_{max} and I_{acc} BEV images that store respectively the minimal elevation, maximal elevation and number of points projected in each pixel. To obtain these

images we use a resolution of 5 *pixels/m* for the xy grid, that is, the size of the pixel side is 20 *cm*. Along with this, the BEV images are 8-bit encoded images and the resolution used for the elevation is 10 *levels-of-gray/m*.

Differently from the original work, we interpolate and segment I_{max} image for a higher confidence in the detected ground. The method can be divided into the following steps:

1. identify the ground around the scanner,
2. build a polar grid and interpolate values,
3. compute λ-flat zones and extract ground on BEV image,
4. back project ground label from BEV image to 3D points.

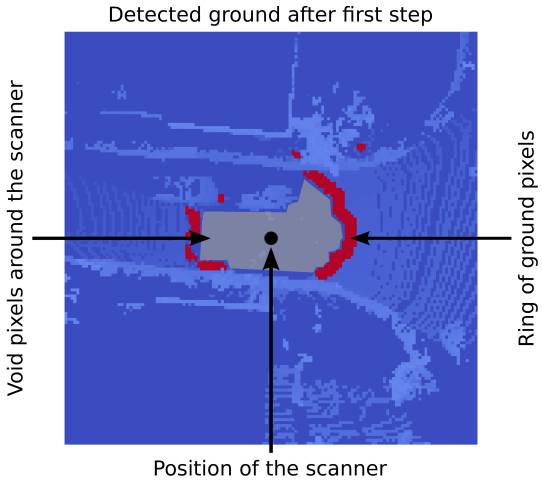


Figure 3. A zoom of the image I_{max}: in a narrow street, the road in front of the car is disconnected from the rear. In red, pixels in the closest ring around the scanner are detected as ground.

3.1 Identify the ground around the scanner

The first step is to retrieve the part of the ground closest to the car. The goal is to reconnect the road in front of the car with the one behind. In the original method, the ground is identified as the biggest λ-flat zone found after segmenting the image. In situations where the car is navigating through narrow streets, this assumption may not be verified, just because the ground in front of the car could not be connected with the ground in the rear, as shown in Figure 3. In the proposed example, pixels in the sides of the car represent either a wall or other cars, the ground in the front is disconnected from the one behind. To solve this problem, we detect the ground among the pixels in the closest ring around the car. These pixels will be used later on as markers to detect which λ-flat zones belong to the ground and merge them together. We start identifying the circle made of void pixels around the scanner using a morphological reconstruction by dilation. We use as marker image f:

$$f(x,y) = \begin{cases} 255 & \textit{if } (x,y) = (x_0, y_0), \\ 0 & \textit{otherwise.} \end{cases}$$

where (x_0, y_0) is the pixel corresponding to the position of the scanner in the image. Furthermore, we use as mask image g:

$$g(x,y) = \begin{cases} 255 & \textit{if } I_{acc}(x,y) = 0, \\ 0 & \textit{otherwise.} \end{cases}$$

Thus, the image I_c containing the identified circle is obtained as $I_c = R_g^{\delta}(f)$. Then, we detect among the points in the closest ring around the car those belonging to the ground. To achieve this, we first locate the ring $\mathcal{R}$ around the car applying a morphological external gradient defined as: $I_r = \delta_B(I_c) - I_c$, where B is a structuring element of size $1m^2$. The ring is the set $\mathcal{R} = \{(x,y) \mid I_r(x,y) = 255\}$. Then we compute

$$z = \min_{(x,y)\in\mathcal{R}} I_{max}(x,y),$$

the smallest z value in I_{max} on the set $\mathcal{R}$. Finally, we assign as ground only the pixels (x, y) in the ring $\mathcal{R}$ such that $|I_{max}(x,y) - z| < 0.5m$. Figure 3 illustrates in red the resulting detected ground.

3.2 Build *Dartboard* and Interpolate image

In the second step, we interpolate information contained in I_{max} image. This is a necessary step in the method because it fills information on void pixels. Namely, we define a polar grid and then we map pixels of I_{max} image onto its elements. To better explain our choice, let us analyze how points are spatially located in an ideal environment where the ground is a plane orthogonal to the axis of the scanner. The scanner spins around its vertical axis. Looking at points for a fixed yaw angle, as in Figure 4, we can see that the distance of the points from the scanner grows with the tangent of φ_i. Thus, in this context, a polar grid on the xy plane, where the length of intervals in the radial axis increases with a tangential trend, would be better suited than the Euclidean grid to prevent disconnections.

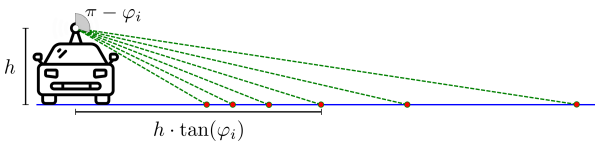


Figure 4. The distance between points and the scanner depend on the tangent of the inclination angle (φ_i) of the layer i and the scanner height h.

To define the intervals in the radial axis, let first consider $l_1, \ldots, l_n$ layers in the scanner, and let $0 \leq \varphi_1 \leq \ldots \leq \varphi_n \leq \frac{\pi}{2}$ their respective inclination angles. Furthermore, let h be the distance between the scanner and the ground. In the hypothesis of an ideal environment, we can estimate the radial distances r_i of the scanned points as:

$$r_i = h \cdot \tan(\varphi_i), \quad \forall i = 1, \ldots, n.$$

Hence, we split the radial axis with intervals $[r_i, r_{i+1})$, for $i = 0, \ldots, n+1$, where $r_0 = 0$, and $r_{n+1} = \infty$. In this way, the profile of the ground in the grid remains connected because for each cell we have at least one point that falls in. Differently from the radial axis, we choose $0 \leq \theta_1 \leq \ldots \leq \theta_m \leq 2\pi$ angle to evenly split the polar axis. Thus, each element $\mathcal{S}$ of the *dartboard* is defined as the set of pixels in the product $[r_i, r_{i+1}) \times [\theta_j, \theta_{j+1})$. Figure 5 shows the *dartboard* obtained for the KITTI Benchmark (Geiger et al., 2012) (scanner height $h = 1.73m$).

Once generated the *dartboard*, it is used to interpolate information on void pixels in I_{max} image and obtain the interpolated image $\hat{I}_{max}$. Then for each pixel x, y in the euclidean grid we

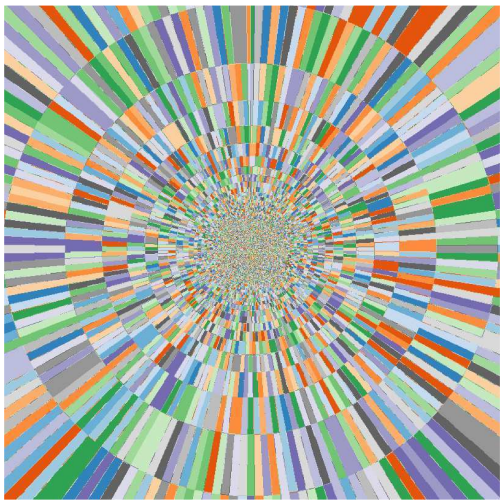


Figure 5. Dartboard: each radial sector is covered by a different scanner layer.

compute its polar coordinates r, θ as:

$$\begin{cases} r = \sqrt{(x - x_0)^2 + (y - y_0)^2}, \\ \theta = arctan((y - y_0)/(x - x_0)). \end{cases}$$

where (x_0, y_0) is the scanner location. The coordinates (r, θ), determine a circular sector $\mathcal{S}$ in the polar grid. In this way, we map each element in the I_{max} domain to an element in the *dartboard*. Now, let us define the image $\hat{I}$ obtained assigning the minimum in each *dartboard* circular sector

$$\hat{I}(x, y) = \min_{(i,j) \in S_k} I_{max}(i, j).$$

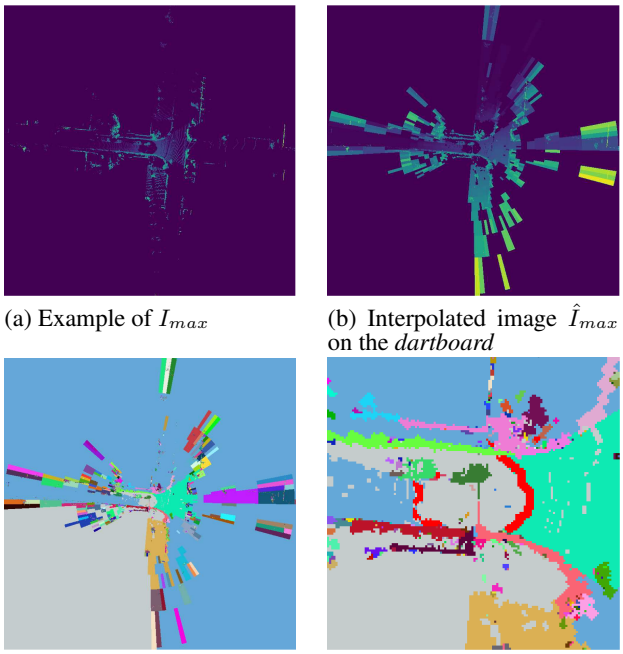
where S_k is the circular sector containing pixel (x, y). Only I_{max} null (empty) pixels are interpolated. Thus, the final interpolated image $\hat{I}_{max}$ is computed as the maximum between I_{max} and $\hat{I}$:

$$\hat{I}_{max}(x, y) = max(I_{max}(x, y), \hat{I}(x, y)),$$

Figure 6 illustrates an interpolation example. Observe how peripheral pixels that were disconnected in I_{max} image (figure 6 (a)) are reconnected in $\hat{I}_{max}$ (figure 6(b)).

3.3 Compute λ-flat zones and extract ground on BEV image

λ-flat zones are computed on $\hat{I}_{max}$ image. Similarly to (Hernández and Marcotegui, 2009), we use $\lambda = 0.20m$. An example of the obtained λ-flat zones is illustrated in Figure 6. Note that in this example, the car is navigating through a narrow street and the road is divided into two main λ-flat zones. To merge the two connected components the marker extracted in section 3.1 is used. Figure 6 (d) shows a zoom of the obtained segmentation around the car. The red ring in the center of the image is the ground marker detected in section 3.1. The detected ground is composed by the union of λ-flat zones whose intersection with the detected ground marker is not empty. At the end of this step, the method returns a binary label BEV image I_g, whose non-zero pixels represent the detected ground.



(a) Example of I_{max}

(b) Interpolated image $\hat{I}_{max}$ on the *dartboard*

(c) Quasi-flat zones with $\lambda = 0.2m$.

(d) Zoom around the car. Red pixels represent the ground detected in section 3.1.

Figure 6. Example of analysis in interpolated image obtained on the frame 3721 in sequence 08 of SemanticKITTI dataset.

3.4 Back project labels

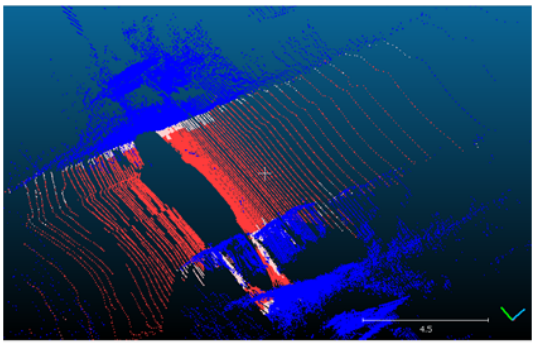
Finally, the detected ground pixels must be projected back to 3D point clouds. As said before, ground is detected on I_{max} image instead of I_{min} image proposed in the original method (Hernández and Marcotegui, 2009). Even though the two approaches seem similar, there is a significant difference between them that has to be considered before the back projection of the labels. Using the I_{max} image, ground points close to objects or under them (such as ground under a tree) are not detected as ground. The reason of this problem is that I_{max} refers to the object elevation and the λ flat zone propagation can not reach it from the ground. An example of this issue is illustrated in figure 7. Red points represent true positive ground points, blue points represent true negative and white points represent false negative points, ground points close to objects. This effect is stronger for lower resolutions. To solve the problem, ground pixels are extended on I_{min} λ-flat zones. First of all we compute λ-flat zones on I_{min} image. Then, let us define $\bar{I}_g$ the extended ground image as:

$$\bar{I}_g(\mathcal{C}) = \max_{(x,y) \in \mathcal{C}} I_g(x, y),$$

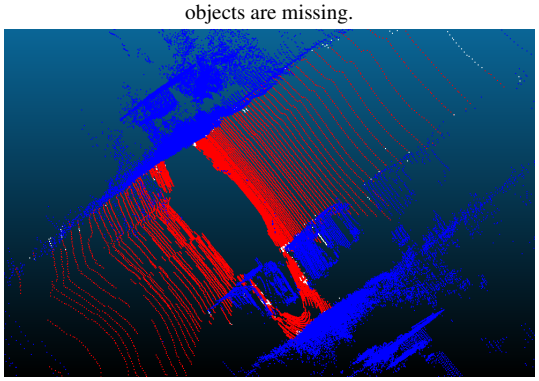
for each quasi flat zone $\mathcal{C}$ obtained from I_{min}. Intuitively, we propagate ground labels on λ-flat zones computed on I_{min} image. In this way, we assign as ground pixels containing both ground points and points belonging to objects. Hence, during the back projection, we need to separate this last group of points. To achieve this, let us consider $\mathcal{F} = \{(x, y) | I_g(x, y) = 1\}$ subset of the image domain made by pixels initially marked as ground, and let us consider $\bar{\mathcal{F}} = \{(x, y) | I_g(x, y) = 0 \wedge \bar{I}_g(x, y) = 1\}$ subset of the image domain made by pixels where the ground label has been extended. Let p be a point and let (x_p, y_p) the pixel in the image domain where p is projected. If $(x_p, y_p) \in \mathcal{F} \cup \bar{\mathcal{F}}$ then the point p may belong to the ground.

To decide if p belongs to the ground or not, we consider the difference between its elevation and the corresponding value in I_{min} image, *i.e.* $|p_z - I_{min}(x_p, y_p)|$. Two different threshold values, respectively $\delta_{\mathcal{F}} = 20\ cm$ and $\delta_{\bar{\mathcal{F}}} = 5\ cm$, are defined according to whether an object has been detected in the pixel initially detected as ground ($\mathcal{F}$) or not ($\bar{\mathcal{F}}$). If so, the tolerance is lower in order to prevent the inclusion of the lower part of the object into the ground. The label $l(p)$ assigned to the point p is:

$$l(p) = \begin{cases} 1 & \textit{if } (x_p, y_p) \in \mathcal{F} \wedge |p_z - I_{min}(x_p, y_p)| \leq \delta_{\mathcal{F}}, \\ 1 & \textit{if } (x_p, y_p) \in \bar{\mathcal{F}} \wedge |p_z - I_{min}(x_p, y_p)| \leq \delta_{\bar{\mathcal{F}}}, \\ 0 & \textit{otherwise.} \end{cases}$$



(a) Ground detected on I_{max}. Ground points close to vertical objects are missing.



(b) Results obtained while including the ground expansion on I_{min} before the back projection.

Figure 7. 3D ground detection results. Red points (TP), blue points (TN) and white points are false negative points.

4. EXPERIMENTS ON GROUND DETECTION

The proposed method is compared against two state of the art algorithms and a naive RANSAC method on the Semantic KITTI dataset (Behley et al., 2019). We include RANSAC in the analysis as a baseline benchmark. Cloth Simulation Filter (CSF) (Zhang et al., 2016) proved great adaptability to a wide range of different environments, either urban and rural. In addition, we use a FCNN method similar to the one proposed in (Velas et al., 2018). The main difference with the original is the network used. Instead of employing the architecture proposed by the authors, we use a U-Net architecture (Ronneberger et al., 2015), for its great versatility to different applications. In order to train and validate the U-Net model, we select one scan over ten in the sequences from 0 to 10 except for the sequence 08. We adopt this last sequence as a test set for all the methods. The split between training and test has been done following directives in (Behley et al., 2019).

Since the dataset does not contain an explicit ground class, we derived it by aggregating multiple classes (Road, Parking, Sidewalk, Other-Ground, Lane-Marking and Terrain). Furthermore, to have an overview of classification errors made in the predictions we created a total of eight categories aggregating all classes. The categories that we created are Ground, Building, Vehicles, Cycles, Person, Vegetation, Fixed-Objects, and Moving Objects.

We use the following metrics to benchmark our experiments, $P = \frac{TP}{TP+FP}$ as *precision*, $R = \frac{TP}{TP+FN}$ as *recall*, $A = \frac{TP+TN}{TP+TN+FP+FN}$ as *accuracy*, and *Intersection over Union* (IoU) also called Jaccard Index $IoU(A, B) = \frac{|A \cup B|}{|A \cap B|}$ where TP, TN, FP, FN indicate respectively the number of true positives, true negatives, false positives and false negatives, and A, B are any two sets. The sets used to compute the Jaccard Index are the set of predictions and the ground truth. In all the cases the scores have been measured using the predictions on the 3D point clouds.

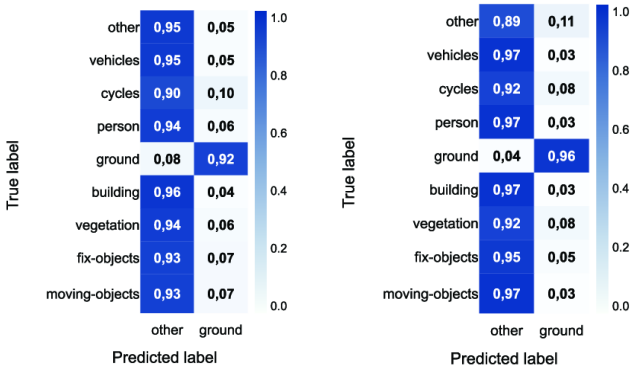
Table 1 lists the scores obtained by the methods. The table is divided in two parts: the first lists the unsupervised methods and the second the supervised approach. All the methods analysed achieve great performances, and the FCNN achieves the highest score in almost all the metrics. Note that our proposed BEV λ-FZ method shows a good trade off between precision and recall, and among the unsupervised methods is the one with the highest Jaccard Index. Moreover, this method needs just a few parameters to work and this makes it much easier to explain why it fails compared to FCNN. Along with these metrics, we analyze the confusion matrix in order to evaluate which categories are confused with the ground . Thus, Figure 8 shows the confusion matrices. The vegetation is the class with the highest rate of points classified as ground. This category contains low plants and separating them from terrain with propagation approaches is cumbersome.

Method	**F_1**	**Recall**	**Precis.**	**Acc.**	**IoU**
RANSAC	.922	.917	.927	.930	.856
CSF	.937	.976	.900	.940	.881
BEV λ-FZ	.945	.960	.930	.949	.895
FCNN	.951	.921	.982	.957	.907

Table 1. Quantitative results obtained on sequence 08 of SemanticKITTI dataset for the ground detection task. CSF (Zhang et al., 2016). FCNN (Velas et al., 2018) BEV λ-FZ is the method proposed in this paper.

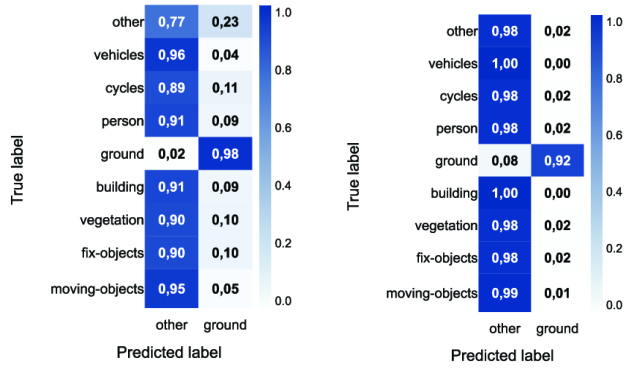
Let now analyse qualitatively the results and see some examples in which our proposed method fails. To visualize the predictions in the following figures we use the color code hereby, green for TP, red for FP, blue for FN and gray for TN.

Figure 9 shows stairs classified as ground by our method (λ value is larger than the step height). CSF and FCNN methods do not prevent completely this error. Figure 10 misses the detection of a garden because it is behind a bush. This bush prevents the garden λ-FZ to reach the ground marker around the car. From an autonomous driving application this is not an issue. It can even be seen as an advantage, as the garden behind the bush is not reachable by the car. Moreover, this kind of missing detection happens only in isolated zones of the scene that cannot be easily reached. This is confirmed also by the high recall rate of the method. Finally Figure 11 shows an example



(a) RANSAC

(b) BEV λ-FZ



(c) CSF

(d) FCNN

Figure 8. The confusion matrices: (a) Naive RANSAC (b) BEV λ-quasi flat zones (c) CSF (d) FCNN based approach.

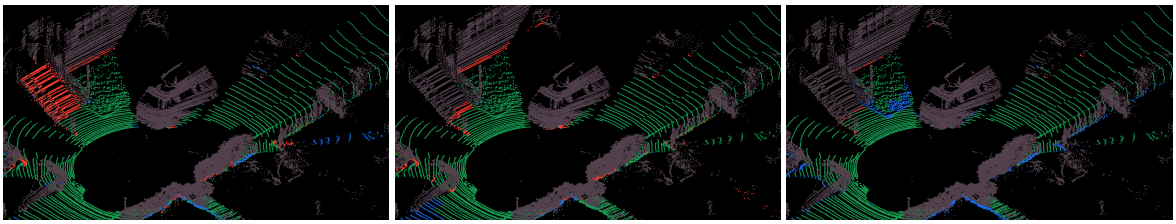
in which our method detects a terrain zone while FCNN method misses it.

5. CONCLUSIONS

In this paper we propose a BEV grid in the form of a dartboard with radial sectors of increasing size with the scanner distance. This grid fits the acquisition system, taking into account the height of the scanner with respect to the ground as well as the number of laser layers and their corresponding inclination angles. The resulting representation is ideal for further processing, avoiding object splitting due to low resolution of faraway objects as well as losing details for nearby objects. The improved BEV representation is a better starting point for any BEV analysis approach. We revisit a simple ground detection method based on the assumption of small elevation variations. We introduce the *dartboard* grid and we demonstrate good performances. We have compared our method with two state of the art methods (CSF and FCNN) and a naive RANSAC on the SemanticKITTI dataset. Results show that our method is comparable with other state of the art algorithms, even though FCNN is more precise. In our opinion, the few parameters used and the greater explicability in case of error compared to FCNN make our algorithm a good candidate for potential applications. Moreover, the proposed BEV representation can also be used in other learning strategies.

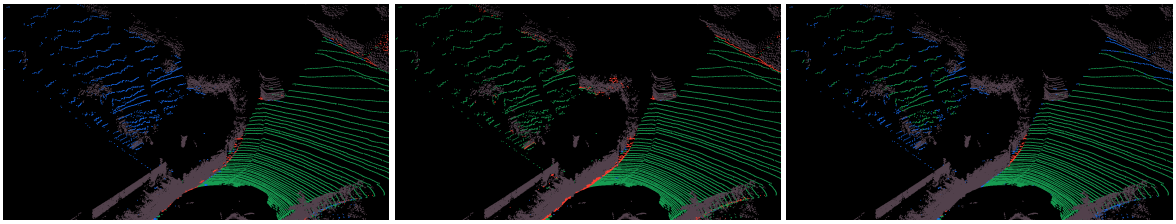
ACKNOWLEDGMENTS

This work was partially funded by REPLICA FUI 24 project and a MINES ParisTech PhD scholarship.



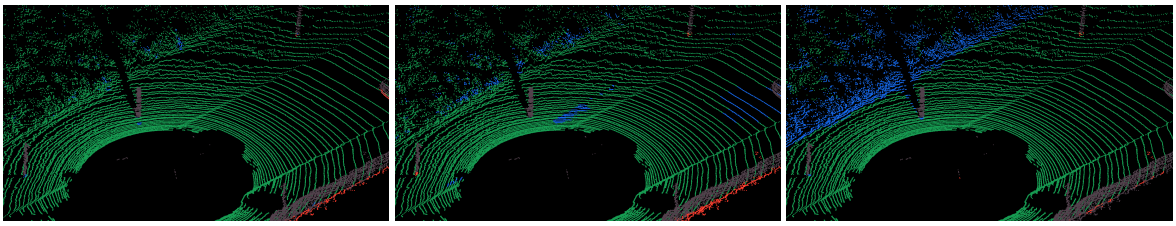
(a) BEV λ-FZ (b) CSF (c) FCNN

Figure 9. Qualitative results. Green points are true positives, red ones are false positives, blue false negatives and gray ones are true negatives. BEV λ-FZ includes stairs nearby the road as ground. The λ used is larger than the step. FCNN misses some nearby ground points.



(a) BEV λ-FZ (b) CSF (c) FCNN

Figure 10. Qualitative results. Green points are true positives, red ones are false positives, blue false negatives and gray ones are true negatives. BEV λ-FZ considers as ground the biggest flat zone in the projection image. In this example a piece of the garden is missing because it is behind a bush.



(a) BEV λ-FZ (b) CSF (c) FCNN

Figure 11. Qualitative results. Green points are true positives, red ones are false positives, blue false negatives and gray ones are true negatives. Predictions obtained by the three analysed methods. In this example FCNN fails to detect some terrain points.

REFERENCES

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J., 2019. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*. 2, 5

Fischler, M. A., Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395. 1

Gallo, O., Manduchi, R., Rafii, A., 2011. CC-RANSAC: Fitting planes in the presence of multiple surfaces in range data. *Pattern Recognition Letters*, 32(3), 403 - 410. http://www.sciencedirect.com/science/article/pii/S0167865510003557. 1

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 3354–3361. 3

Hernández, J., Marcotegui, B., 2009. Point cloud segmentation towards urban ground modeling. *2009 Joint Urban Remote Sensing Event*, IEEE, 1–5. 1, 2, 4

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2020. Randla-net: Efficient semantic segmentation of large-scale point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11108–11117. 2

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4558–4567. 2

Meyer, F., 2001. An Overview of Morphological Segmentation. *Intern. Journal of Pattern Recognition and Artificial Intelligence*, 15(7), 1089-1118. 2

Milioto, A., Vizzo, I., Behley, J., Stachniss, C., 2019. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2

Oniga, F., Nedevschi, S., Meinecke, M. M., To, T. B., 2007. Road surface and obstacle detection based on elevation maps from dense stereo. *2007 IEEE Intelligent Transportation Systems Conference*, IEEE, 859–865. 1

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241. 5

Roynard, X., Deschaud, J.-E., Goulette, F., 2016. Fast and robust segmentation and classification for change detection in urban point clouds. *ISPRS 2016-XXIII ISPRS Congress*. 1

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient ransac for point-cloud shape detection. *Computer graphics forum*, 26, Wiley Online Library, 214–226. 1

Serna, A., Marcotegui, B., 2013. Urban accessibility diagnosis from mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 84, 23–32. 1

Serna, A., Marcotegui, B., 2014. Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 243–255. 1

Soille, P., 2008. Constrained connectivity for hierarchical image partitioning and simplification. *IEEE transactions on pattern analysis and machine intelligence*, 30(7), 1132–1145. 2

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L. J., 2019. Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6411–6420. 2

Velas, M., Spanel, M., Hradis, M., Herout, A., 2018. CNN for very fast ground segmentation in velodyne lidar data. *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, IEEE, 97–103. 2, 5

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sensing*, 8(6), 501. 1, 5

Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., Foroosh, H., 2020. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9601–9610. 2

Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., Lin, D., 2021. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9939–9948. 2