Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Drivable Area Auto-labelling Based on Elevation Map and Clustering

Chengrui Zhu†*1, Junyuan Lu†*1, Qimeng Tan2, Yue Wang‡1
[1]*College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China*
[2]*Beijing Institute of Spacecraft System Engineering CAST, Beijing 100076, China*
†E-mail: jewelry.zju@gmail.com; junyl@zju.edu.cn

Abstract: Environment sensoring is always a limelight in automatic driving. We propose a novel drivable area auto-labelling method based on LiDAR and monocular camera, which constructs a elevation map, calculates drivablity of each grid and gets continuous drivable areas via clustering and vertical growing. Finally, We prove the effectiveness of our labels with a classic network training framework on KITTI dataset.

Key words: Rrivable area; Automatically labelling; Elevation map; Clustering
https://doi.org/ **CLC number:** TP391.4

1 Introduction

Real-time sensing of local environment is the basis of autonomous navigation of robot. Elevation map is a common description of local environment, which can be constructed with a variety of robot sensors and updated with the change of the position and pose of a robot. However, the performance of existing methods of constructing elevation map based on vision have many redundant intermediate processes. Our research contributes to accelerating it through deep learning, improving the efficiency and performance of drivable area detection.

In recent years, computer vision has been booming with the development of deep neural network. Many network frameworks, such as FCN(Long et al., 2015) and U-Net(Ronneberger et al., 2015), relying on numerous accurate labelled data, have made significant progress in the semantic segmentation of road drivable areas. However, collecting data and labelling is rather time-consuming and laborious. To tackle this problem, we propose a low-cost and accurate method for constructing effective training datasets, named *Drivable Area Label Generation Based on Elevation Map and Clustering.*

Our method consists of two parts. For the first part, it constructs elevation maps by point clouds from LiDAR, and the drivability of each grid is calculated by both region growing based on height, the surface normal and neighbor height deviation. Then, it projects drivability information to the corresponding images through intrinsics and extrinsics of the camera. For the second part, it uses hierarchical clustering for significantly undrivable areas, and vertical growing for significantly drivable areas. Finally the experimental results proved its effectiveness for improving the network performance.

2 Related works

In the field of drivable area detection, there have been many effective methods, including traditional, supervised and semi-supervised ones. These methods can also be divided into methods relying only on

[*] Joint first authors
[‡] Corresponding author
ORCID: Chengrui Zhu, https://orcid.org/0000-0002-6382-6569

vision, only on LiDAR or combining both. Some of the camera-based methods rely on global road priori hypothesis, such as road boundaries(Yuan et al., 2015)(Aly, 2008), lane lines, and vanishing points (Kong et al., 2010)(Alvarez et al., 2014). However, road conditions are extremely complicated and somtimes violate those priori hypothesis, which undermines their portablity. In addition, visual information is easily influenced by light conditions, and vision-based algorithms almost loses efficiency in dark environments. A Laser-based algorithm (Caltagirone et al., 2017) uses the top view of the LiDAR point cloud to train a FCN network to acquire a depth map, but the sparsity of LiDAR also limits the effect of the method. (Caltagirone et al., 2019) uses LiDAR and visual images to train a FCN network. By cross-fusing point clouds and corresponding RGB images in the training process, network learns more abundant information and gets a better result on segmentation. However, a large number of manual labelled images are still required in the training of the network.

Therefore, automatically labelling methods are introduced to training segmentic segmentation networks. (Laddha et al., 2016) uses OpenStreetMap and location sensors to cooperate in drivable area labelling, and detects dynamic obstacles on a actual road by training a CNN network, ineffective in lack of accurate map data. (Barnes et al., 2017) proposes a weak-supervised training method, which uses a visual odometer to perceive the forward path, projects LiDAR data onto the image and uses a top-down search method to get the obstacle border, with poor detection efficiency in the scene of road intersection. In (Gao et al., 2019), 3D LiDAR data, the real trajectory from GPS, and a special network framework is designed for the ambiguous areas on off-road roads, where accurate GPS positioning necessitates. (Pan et al., 2021) proposes an elevation mapping system and integrates traversability analysis into simultaneous localization and mapping (SLAM), which decreases the computation cost of obstacle awareness.

3 Methodology

3.1 Grid drivability calculation

In this section, we construct a elevation map and calculate the drivability of each grid of that.

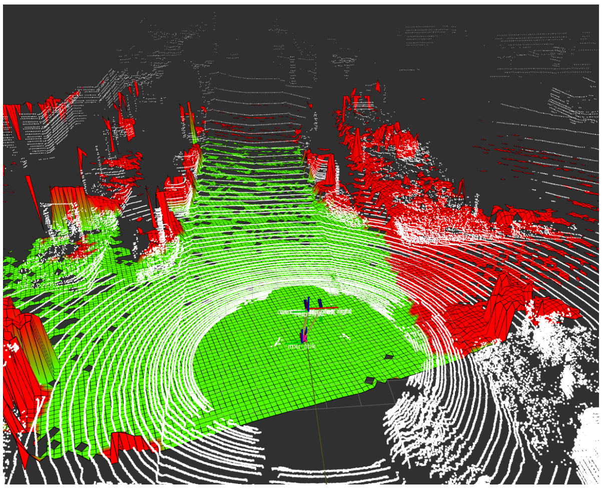

Fig. 1 Region growing in 3D point of view, where drivable grids are marked green.

First, GPU Accelerated Elevation Map Construction (Pan et al., 2019) is adopted to construct a local elevation map, where surface normal and neighbor height deviation are calculated according to equation 1 of each grid:

$$\begin{aligned} \boldsymbol{n}_i &= \min_x ||(\boldsymbol{Q}_i - \mathbf{1}_k \boldsymbol{p}_i^{\mathrm{T}}) n_i||_2 \\ H_d &= |h_{P_{(x,y)}} - \bar{h}| \end{aligned} \tag{1}$$

A novel approach though it is, it has a poor detection effect on certain obstacles such as lawn and steps at the sides of the road, thus the drivable areas are arbitrary to some extent. Above method will be called the control method below, and on the basis of this, we introduce region growing to generate the drivable area, in order to obtain conservative drivable areas.

Region growing(Adams and Bischof, 1994) is a method of aggregating pixels and subregions into larger regions according to predefined criterion. It starts from a group of seed points, adds their neighbor regions with similar properties to the next generation of seed points, and ends if seed points can't expand any more. In our method, the height of each grid is regarded as the criteria of growing.

In our method, it's assumed that the current position of the car is always drivable, so we set current position as initial seeds, and set the growing threshold to 0.05m. By this way a more conservative drivability judgement is made compared with the control method, meanwhile capable to accurately detect grassland, small steps, etc., which is clearly shown in Fig. 1, Moreover, due to the significant

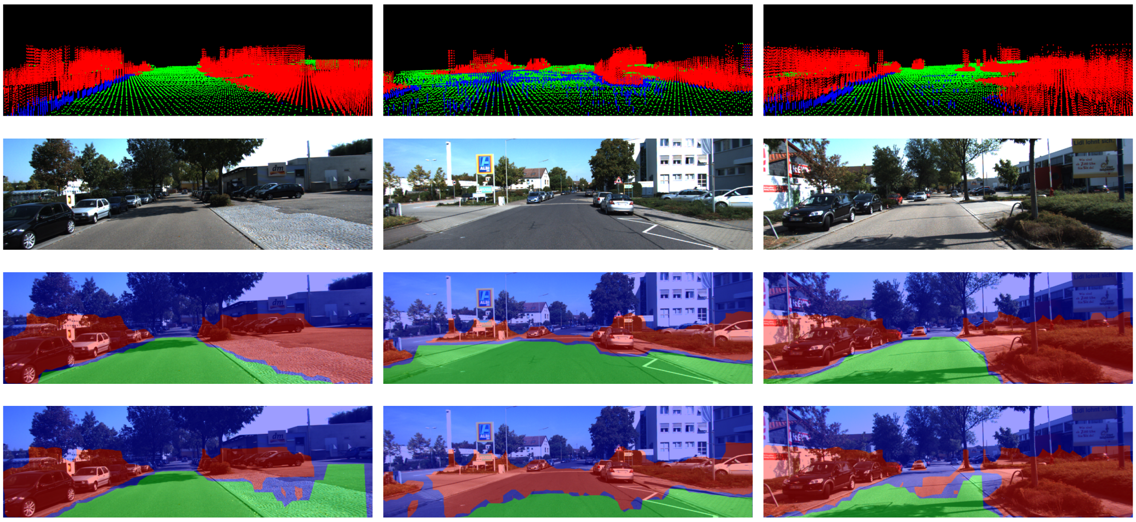

Fig. 2 The first row is the projected points with drivability, the second is the original visual image, the third uses both methods, while the fourth only uses a complex of surface normal and neighbor height deviation. In the images, green, red and blue denotes drivable, undrivable and ambiguous separately. It can be seen that region growing and KNN significantly improve the quality of labels.

noise of LiDAR height measurement in the distance which probably exceeds the threshold, region growing is limited to the vicinity. In view of this, a flexible threshold is set with respect to the distance to the area.

To take advantage of the characteristic of surface normal and neighbor height deviation, which is obstacle perception in advance brought out by differential prediction effect, we fuse the two methods by the following strategy. For every grid:

- if drivable for both methods, it's drivable;
- if undrivable for region growing, it's undrivable;
- if undrivable for the weighted method but drivable for region growing, it will finally be determined by KNN (k-nearest neighbor).

3.2 Projective transformation

In this section, drivability information is projected from the elevation map to the 2D image plane. Given coordinates of all grids on elevation maps in the map frame and their drivability, via the transformation matrix from the map frame to the sensor frame, drivability is projected to the LiDAR coordinate system. With intrinsics, extrinsics and distortion coefficients, drivability is shown as discrete points in Fig. 2. Equation 2 demostrates the projection relations:

$$\begin{aligned} \boldsymbol{X} &= [x, y, z, 1]^{\mathrm{T}}, \quad \boldsymbol{Y} = [x, y, 1]^{\mathrm{T}} \\ \boldsymbol{Y} &= \boldsymbol{P}_{rect} \cdot \boldsymbol{R}_{rect} \cdot (\boldsymbol{R}|\boldsymbol{T})_{velo}^{cam} \cdot \boldsymbol{X} \end{aligned} \tag{2}$$

where $\boldsymbol{X}$ denotes 3D homogeneous coordinates, $\boldsymbol{Y}$ denotes 2D homogeneous coordinates on the image plane, $(\boldsymbol{R}|\boldsymbol{T})_{velo}^{cam}$ denotes the transformation matrix from velodyne to camera, $\boldsymbol{R}_{rect}$ denotes the distortion rectification matrix and $\boldsymbol{P}_{rect}$ denotes the projection matrix.

3.3 Area aggregation

In this section, we aggregate discrete drivability projections into continuous drivable or undrivable areas, thus append a pixel-level drivability label layer to a visual image, which can be called drivability image. A typical drivability image generated by our method contains three types of labels: significantly drivable, significantly undrivable and ambiguous. The ambiguous areas are somewhere hard to judge its drivablity, which typically lies in the vicinity of the edges of drivable and undrivable areas. In the network training process, the ambiguous areas are ignored instead of regarded as the third class.

In typical urban road environments, there are obstacles of various shapes around the vehicle, the sum of which is the so-called undrivable area. Given the undrivable projections, our solution is to cluster all the projections and then extract the boundary of

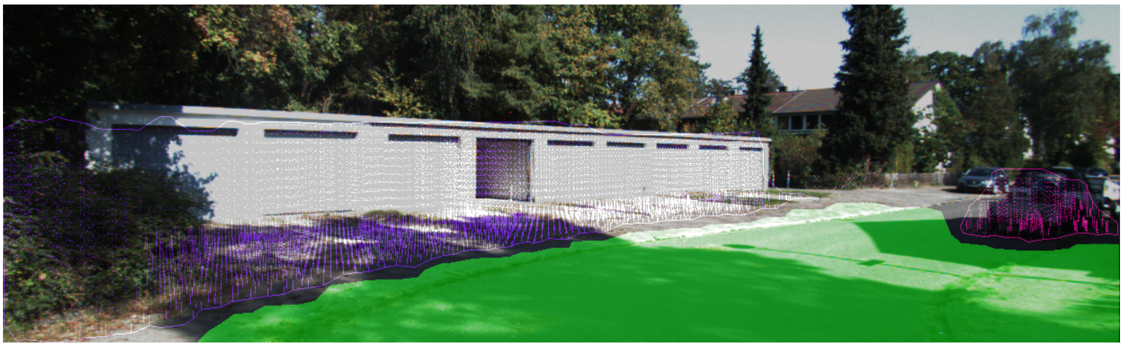

Fig. 3 Clustering and vertical growing result instance, in which obstacles are divided into two clusters: the architecture (in purple) and the car (in red) and encircled separately.

each cluster separately. Ideally, the points of each cluster belong to different obstacles, the outlines of which are partitioned by the corresponding boundary of projections.

Therefore, we combine Hierarchical Clustering (Johnson, 1967) and the circumscribed polygon boundary extraction. Compared with other clustering methods such as K-means and Gaussian mixture models etc., which mainly focus on inner-class similarities and inter-class discrepancies, hierarchical clustering can flexibly select the clustering criteria to cater to various clustering purposes. In drivable area segmentation, we concentrate more on the continuity of projections. Therefore, the inter-class distance is defined as the shortest distance between two points respectively belonging to two clusters, i.e.

$$\mathrm{Dist}(X,Y) = \min_{x\in X, y\in Y} \mathrm{dist}(x,y) \tag{3}$$

where X and Y denote two clusters; x and y denote two points respectively belonging to X and Y. This clustering method is also called Single-linkage Agglomerative Clustering (Sibson, 1973). Then Delaunay triangulation(Zhigeng et al., 1996) is adopted to extract the circumscribed polygon of projections for each cluster, and the areas within the polygons is regarded undrivable. A cluster with few projections inside can be filtered because it's presumably a noise spot.

Compared with the acquisition of the undrivable areas, a more conservative method *vertical growing* method for drivable areas. A distinct feature of drivable areas is that they always extends from the current position to the nearest obstacle, and when reflected on visual images, they grow vertically from the bottom to the farthest drivable projection below undrivable projections. The drivable areas generated in this way never overlap with the undrivable areas.

In practice, we divide the image vertically into several columns, and adopt vertical growing for each column. Specifically, for each column of images, we traverse the projections in order from bottom to top, and stops if two neighbor projections are far apart (not continuous) or it's too close to the undrivable area (not significant). Thus, we have acquired a sequence of the pixel heights of the drivable areas in each column of images and adopt filtering and interpolation if necessary (median filtering and linear interpolation in the experiment). In addition, we truncates the drivable area where the pixel height is trivial. Finally, the continuous boundaries of the significantly drivable areas are acquired.

In our method, lots of restrictions are impose to the significant drivable area, which ensures its continuity and accuracy. That contributes to improvement of precision but meanwhile deterioration of recall. As precision and recall are usually seen as a trade-off and precision is much more significant than recall in this case, it's reasonable to sacrifice recall appropriately.

4 Experiments

4.1 Network training

'With the control method and our method separately, we generate two group of datasets with the image number of 56, 112, 224 and 336 as training set. Moreover, KITTI Pixel-level Semantic Segmentation Benchmark(Geiger et al., 2012) is splitted into 112, 60 and 28 images which is added respectively to training, validation and test sets. It's worth mention-

Table 1 Experiment result comparision

Generated	**Control group(%)**				**Our Method(%)**			
Label	*Manual Selection of Basic Labelling*				*Automatic Selection with Improvements*			
Number	Precision	Recall	Accuracy	IoU	Precision	Recall	Accuracy	IoU
0	80.5	**93.1**	93.3	76.0	80.5	**93.1**	93.3	76.0
56	87.2	87.4	94.2	77.4	84.0	90.1	93.8	76.9
112	**89.4**	87.0	**94.7**	**78.9**	**88.1**	88.0	**94.6**	**78.6**
224	89.3	86.7	94.6	78.6	86.2	88.9	94.2	77.8
336	89.2	85.1	94.3	77.2	85.6	87.0	93.7	75.9

ing that the datasets generated the control method is manually selected because most of the labels are rough and erroneous.

Based on KITTI Raw Data, we can generate far more datasets than we need, so a fairly stringent filtering strategy is made to ensure the high quality of the labels. In practice, we set the upper limit of ambiguous areas and the lower limit of drivable and undrivable areas. Since only significant labelled areas are utilized during training, the generated dataset has only a good effect on the neural network if the correctness of the vast majority of labels is ensured. This explains why precision is fairly important.

We use the classic FCN8s neural network and commonly used image augmentation methods for training, such as random horizontal flipping, random resizing and cropping, color jittering, etc.; and use the Adam optimizer with weight decay for optimization. Then select the batch which performs best in the validation set for testing, and finally compare the experimental results.

In order to evaluate the performance of the network based on different datasets, the following 4 quantitative indicators are calculated: precision, recall, accuracy and IoU (Intersection over Union). The results are shown in Table 1.

4.2 Results and analysis

Based on a huge amount of data, after manual selection (3000 images per hour), datasets generated from the control method generally has a positive effect on training. The IoU of test set increases first and then decreases with the increasing number of generated data added. In addition, because the drivable area labelling of the manually selected generated data set is more conservative than accurate labels, accuracy increases significantly, while the recall rate decreases slightly. This shows that the automatic labelling algorithm alleviates network's dependence on accurate manually labelled datasets, but generates poor labels in average and still requires manual selection.

In the combined dataset fully automatically generated by the improved method, the indicators varies similarly to control group with respect to the amount of added generated dataset. And the optimal result of IoU (78.6) can be comparable to that of control group (78.9). As expected, the recall of the test results decreased less because of fewer label errors, yet the accuracy improved less correspondingly. The result demostrates that the improved method can generate labels whose style similar to that of manual labels, meanwhile greatly reducing the cost of manual labelling and improves the segmentation performance of the network based on insufficient accurate labels.

5 Conclusion

We propose a drivable area automatically labelling road drivablity labels based on elevation map from LiDAR point clouds, clustering and vertical growing. As a further improvement, we design a label filtering method to improve the quality of labels. Through network training, the effectiveness of our method is proved. Compared with manually labelled datasets, our method is completely automatic and has a negligible cost. For further work, we will further research the influence of noisy labels on training, to utilize auto-generated labels more wisely.

References

Adams R, Bischof L, 1994. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6):641-647.

Alvarez JM, López AM, Gevers T, et al., 2014. Combining priors, appearance, and context for road detection.

IEEE Transactions on Intelligent Transportation Systems, 15(3):1168-1178.

Aly M, 2008. Real time detection of lane markers in urban streets. 2008 IEEE Intelligent Vehicles Symposium, p.7-12.

Barnes D, Maddern W, Posner I, 2017. Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy. 2017 IEEE International Conference on Robotics and Automation (ICRA), p.203-210.

Caltagirone L, Scheidegger S, Svensson L, et al., 2017. Fast lidar-based road detection using fully convolutional neural networks. 2017 ieee intelligent vehicles symposium (iv), p.1019-1024.

Caltagirone L, Bellone M, Svensson L, et al., 2019. Lidar–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125-131.

Gao B, Xu A, Pan Y, et al., 2019. Off-road drivable area extraction using 3d lidar data. 2019 IEEE Intelligent Vehicles Symposium (IV), p.1505-1511.

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, p.3354-3361. https://doi.org/10.1109/CVPR.2012.6248074

Johnson SC, 1967. Hierarchical clustering schemes. *Psychometrika*, 32(3):241-254.

Kong H, Audibert JY, Ponce J, 2010. General road detection from a single image. *IEEE Transactions on Image Processing*, 19(8):2211-2220.

Laddha A, Kocamaz MK, Navarro-Serment LE, et al., 2016. Map-supervised road detection. 2016 IEEE Intelligent Vehicles Symposium (IV), p.118-123.

Long J, Shelhamer E, Darrell T, 2015. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, p.3431-3440.

Pan Y, Xu X, Wang Y, et al., 2019. Gpu accelerated real-time traversability mapping. 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), p.734-740.

Pan Y, Xu X, Ding X, et al., 2021. Gem: Online globally consistent dense elevation mapping for unstructured terrain. *IEEE Transactions on Instrumentation and Measurement*, 70:1-13. https://doi.org/10.1109/TIM.2020.3044338

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, p.234-241.

Sibson R, 1973. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1):30-34.

Yuan Y, Jiang Z, Wang Q, 2015. Video-based road detection via online structural learning. *Neurocomputing*, 168:336-347.

Zhigeng P, Xiaohu M, Jun D, et al., 1996. A graph—based algorlthm for generatlng the delaunay triangulation of a point set within an arbitrary 2d domain (in chinese). *Journal of software*, 007(011):656-661.