2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Fusion of LiDAR and Camera by Scanning in LiDAR Imagery and Image-Guided Diffusion for Urban Road Detection

Yigong Zhang[1], Shuo Gu[1], Jian Yang[1], Jose M. Alvarez[2], Hui Kong[1†]

Abstract— **This paper proposes a new method for road detection based on a 3D LiDAR and a camera. First, the original LiDAR point cloud is re-organized in an ordered way to generate a LiDAR imagery. Then the flat region is extracted from the LiDAR imagery as the candidate road region. Next, a strategy of row- and column- scanning is given in the LiDAR imagery to detect a finer road region from the candidate region. To fuse the point cloud with image information, we transform the point cloud that corresponds to the above detected road region to the image space according to the calibration parameters between the LiDAR and camera. Then we give two image-guided diffusion schemes to conduct image segmentation of road area, respectively. Our experiments demonstrate that this training free approach detects the road region fast, accurately and robustly, and compares favorably with the state-of-the-art on the KITTI benchmark.**

I. INTRODUCTION

Due to various road scenes, complex obstacles, different weather conditions and illumination variation, road detection is still a challenging topic in computer vision. To solve these problems, researchers have proposed some methods based on sensor fusion recently, for example, by the combination of camera and 3D Light Detection and Ranging (LiDAR). Most of these methods (e.g. [1], [2]) adopt a quite similar structure: (1) Extracting features in the image and training a classifier for the road region, (2) Using simple geometry information to get a rough road region, or extracting features in the point cloud and training a classifier for the road region, (3) Using a probabilistic graphical model to fuse the results from the camera and LiDAR. However, there exist some drawbacks in these methods: (1) The segmentation or classification method for point cloud could not provide a good result for road detection when it is used alone. (2) The fusion process is usually dominated by image information, and the fusion may fail in a bad illumination condition. (3) In the fusion process, the probabilistic graphical model only considers the intensity difference between the neighbor pixels in the image, neglecting the structured information, such as edge, gradient and texture.

In this paper, we propose a LiDAR-camera based road detection method. First, the point cloud is re-organized in an ordered way to generate a LiDAR imagery. Then the flat region is extracted from the LiDAR imagery as the candidate road region. Next, a strategy of row- and column-scanning is given in the LiDAR imagery to detect a finer road region from the candidate region. To fuse the point cloud with image information, we transform the point cloud that corresponds to the above detected road region to the image space according to the calibration parameters between the LiDAR and camera. Then we give two image-guided diffusion schemes to conduct image segmentation of road area, respectively. Both of them can preserve the sharp boundary edge and texture information. Experiments demonstrate that our framework compares favorably with the-state-of-the-art on the URBAN_ROAD datasets from the KITTI benchmark.

The rest of the paper is organized as follows: We review some related works in Section II, and give the details of our approach in Section III. The experimental results and analyses are presented in Section IV and the conclusions are drawn in Section V. The supplemental results are uploaded to the Github site https://github.com/kiki0378/Scanning.git.

II. RELATED WORKS

In the literature, most camera-based road detection methods rely on color and texture information to train a classifier to determine whether a pixel or super-pixel belongs to road or not. The common classifiers include boosting [3], random forest [1], [2], etc. In recent years, more and more convolutional neural network based methods have been proposed to detect road region and achieved very promising results. This kind of methods can train detectors for an end-to-end segmentation and provide a pixel-wise prediction result. The representative works include Up-Conv [4], MultiNet [5], FCN-LC [6] and DNN [7] etc.

Among the LiDAR-based methods, due to the high precision of LiDAR sensors, some use the regression method to fit the road region. Asvadi et al. propose a piecewise surface fitting algorithm to fit the road plane [8]. Hu et al. [9] also obtain the road plane by fitting the points with the lowest elevation to a 3D plane model. Chen et al. [10] detect and fit the road-side region to segment the ground. Chen et al. [11] fit a straight line in their proposed LiDAR-histogram, like the operation of the v-disparity map. In [12], the unstructured point cloud is projected to the top view

[1] Yigong Zhang, Shuo Gu, Jian Yang and Hui Kong are with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu province, China 210094. E-mail: {zhangyigong0378, gs110817}@gmail.com, {csjyang,konghui}@njust.edu.cn

[2] Jose M. Alvarez is with the Intelligent Vehicle Department, Toyota Research, CA, USA. E-mail: jalvarez.research@gmail.com

† The author responsible for the correspondence of this paper

* The work of Jian Yang was supported by the National Science Fund of China under Grant Nos. U1713208 and 61472187, the 973 Program No.2014CB349303, and Program for Changjiang Scholars.

** The work of Hui Kong was supported in part by the Jiangsu Province Natural Science Foundation under Grant BK20151491 and in part by the Natural Science Foundation of China under Grant 61672287.

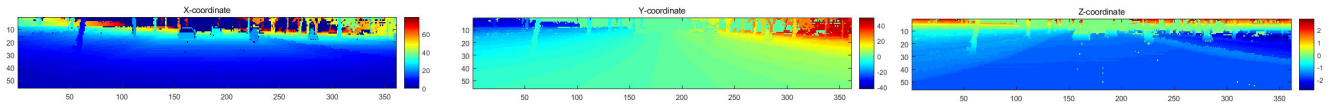

Fig. 1: An example of the LiDAR imagery.

space and the road region is segmented in the top view by the designed convolutional neural network [12].

To utilize both 2D and 3D information from camera and LiDAR, respectively, LiDAR-camera fusion is a natural solution and has attracted more and more attention. In [13], the point cloud is projected to the image space at first and a graph is constructed by Delaunay Triangulation. Then each node in the graph are classified into obstacle or non-obstacle. At last multiple free space is generated to estimate the dense road region. In [9], the fitted road plane is projected into the image plane. Then a Gaussian model of the illumination invariant image feature is learned based on these projected points. Liu et al. [14] conduct the LiDAR-camera fusion to obtain the initial drivable area by a "direction ray map". Then the area is fused with super-pixel, and the fused feature are extracted. Finally, the fusion problem is formulated as a Markov network. In [1], the LiDAR and monocular camera are fused in the framework of conditional random field (CRF) to detect road robustly in different scenes. The two unary potential terms are the classification scores based on the image feature and the 3D point feature, respectively, while the pairwise potential term only considers the difference of the image intensity between two neighbor pixels. Afterwards they further consider the relationship between 3D points and the one between 3D points and image pixels in [2].

The most related works are [11], [13], [14] and [15]. Chen et al [11] define the so-called LiDAR imagery to organize the point cloud from LiDAR. Based on LiDAR imagery and the derived LiDAR histogram, they convert the problem of detecting traversable road and obstacles into a simple linear classification task in 2D space. Both [13] and [14] apply a scanning strategy to find road region. They scan the 3D point in the radial direction in the image space and the triangulation is conducted at the beginning of the algorithms. In contrast, our method only does a row and column scanning in the LiDAR imagery and the fusion operation is done at last. In [15], the 3D points are projected to the image space and generate a dense map by interpolation, then the row and column scanning is conducted. However, it takes a lot of time in the interpolation step, and the coordinates of the 3D points by interpolation are unreliable. Moreover, no image information is used at all. Our method avoids the time-consuming interpolation step and directly operates on the reorganized point cloud – the "LiDAR imagery". Thus it improves the reliability and saves the running time. Meanwhile, the image guided diffusing strategy is adopted to ensure that the boundary of the detected road region via LiDAR imagery is fitted to the one in the image space.

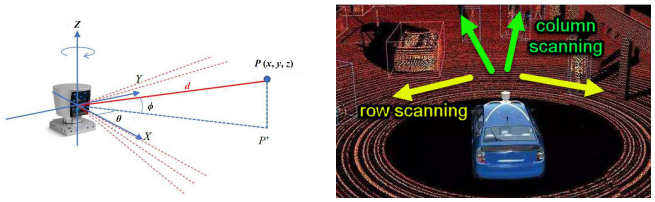

Fig. 2: Left: the coordinate system of a LiDAR sensor. Right: A sketch map for the row- and column-scanning

III. Our Method

A. The LiDAR Imagery

Fig. 2 shows a 3D LiDAR sensor used for the autonomous driving vehicles. The sensor has n lasers aligned vertically to provide a large vertical Field of View (FOV). By self-rotating periodically, it is able to provide a 360° Horizon Field of View (HFOV) with $m-$times sampling. Within a sphere coordinate system, the raw output data of the LiDAR contain three parts: the rotation angle θ, the measurement distance d and the signal strength S. The pitch angle ϕ of each laser can be obtained from the calibration file of the LiDAR. Thus we need to transform the point cloud from the spherical coordinate system (θ, ϕ, d) to the Cartesian coordinate system (x, y, z) by

$$x = d \cdot \cos\phi \cdot \cos\theta, y = d \cdot \cos\phi \cdot \sin\theta, z = d \cdot \sin\phi. \quad (1)$$

Consequently, we obtain a so-called "LiDAR imagery" [11] with a dimension of $n \times m \times 3$. In a LiDAR imagery, both the rotation angle θ (the column number) and the pitch angle ϕ (the row number) are sorted in an ascending order. Each pixel on the same row is obtained by the same laser and shares the same pitch angle, while each pixel in the same column is obtained at the same time and shares the same rotation angle. Each pixel includes a triple - the Cartesian coordinates of a point. Fig. 1 gives an example of the LiDAR imagery. Note that some pixels contain nothing because of no laser reflection. With the effective and efficient representation of the LiDAR imagery, the point cloud is easily indexed and stored in a clear way, which is helpful for searching the road region by a scanning based method.

B. The Flat Region Detection

To reduce the space for the scanning step and remove obstacles, we first detect the flat region in the LiDAR imagery. Let $P_0(x_0, y_0, z_0)$ denote the central pixel and $P_i(x_i, y_i, z_i)(i = 1, 2, \cdots, 8)$ be the eight neighbor pixels of P_0. We hope that all the neighbor pixels can meet the following condition: the height gradient $|z_i - z_0|/\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$ is small enough that the central pixel and its eight neighbors are seen to lie on a flat

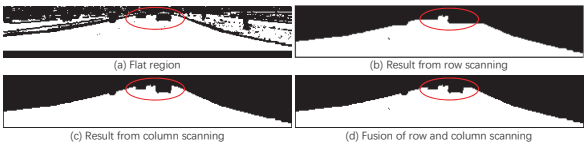

Fig. 3: Examples of the flat region, the row scanning, the column scanning and the fusion result. We can find that the region near the vehicle (in the red circle) is lost when doing the row scanning only. This region is recovered by fusing both the row- and column-scanning.

region, which is similar to [13]. However, if we suppose that the height difference is constant across different rows, the pixel in the bottom row (near the LiDAR) and the one in the top row (far from the LiDAR) may result in different classification results, because the distance in the XY plane between two neighbor pixels $d_{xy}(P_0, P_i) = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$ changes drastically based on their positions. If P_0 is near the LiDAR, $d_{xy}(P_0, P_i)$ is smaller and the height gradient is larger, which may classify the pixel P_0 into the non-flat region. On the other way, if P_0 is far from the LiDAR, $d_{xy}(P_0, P_i)$ is larger and the height gradient is smaller, which may classify the pixel P_0 into the flat region. To solve this problem, we design a new criterion to reduce the influence by the pixel position.

$$flat(P_0) = \sum_{i=1}^{8} \begin{cases} 1, \text{if } \frac{|z_i - z_0|}{dist_i} < \gamma \\ 0, else \end{cases} \tag{2}$$

where $dist_i = \min(\alpha, \max(\beta, d_{xy}(P_0, P_i)))$ makes the height gradient robust to the position of P_0. If the value of $flat(P_0)$ is equal to the number of neighbor pixels with a valid 3D position, the pixel P_0 is classified into the flat region. An example of flat region is shown in Fig. 3(a).

C. The Row- and Column- Scanning

Since the detected flat region is still rough, we perform a row- and column- scanning to refine the road boundary. Both the row- and column-scanning operate in the LiDAR imagery. Fig. 2 gives a sketch map for both the row- and column-scanning. Note that all the operations are executed in the above detected flat region.

1) The Row Scanning: The row scanning is first performed in the LiDAR imagery. It includes two steps: updating the reference pixel and determining the road boundary pixel. In the row scanning procedure, whether a pixel belongs to the road boundary or not is only determined by the height difference between the pixel itself and the reference pixel. Thus, the selection of reference pixels has a large influence on road detection. The first reference pixel is set to the middle pixel of the bottom row in the LiDAR imagery (namely the pixel at the intersection of the $n-$th row and the $\frac{m}{2}$-th column). In general, this pixel always belongs to the road region. Next, we scan its neighbor pixels in the same row to the left and right directions, respectively. Eq.3 gives a criterion to determine whether the current pixel belongs to

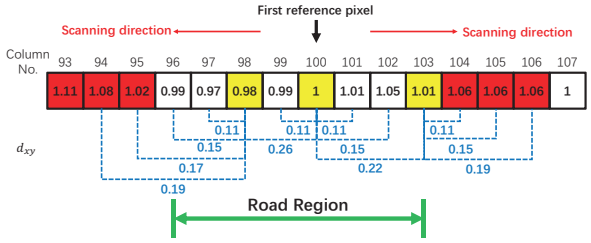

Fig. 4: A sketch map for row scanning. The number above the block represents the column number. The number in the block is the height. The number under the block is the distance in the XY plane d_{xy}. The block filled with yellow represents the reference pixel and the block filled with red represents non-road region. In this case, the first reference pixel is located at the 100-th column, $\Delta_{row} = 0.02$, $\delta_{row} = 0.2$ and the terminal condition for scanning a row is that there exists three consecutive non-road pixels. Note that the block in the 102-th column belongs to the road region because it fails to meet the condition of consecutive non-road pixels. At Last, the road region is located between the 96-th column and the 103-th column. Thus the first reference pixel in the next row is located at the column of $(96 + 103 + 100)/3 = 99$.

the road region or not,

$$P_C \begin{cases} \in \text{road region}, & \text{if } \|z_R - z_C\| > \Delta_{row} \\ \notin \text{road region}, & \text{otherwise} \end{cases} \tag{3}$$

where the height of the reference pixel P_R and the current pixel P_C is represented by z_R and z_C, respectively.

If the current pixel is located in the road region, based on the criterion in Eq.4, we proceed to determine whether we should update the current pixel as the new reference point or not,

$$P_C \begin{cases} \text{new reference pixel}, & \text{if } d_{xy}(P_R, P_C) \geq \delta_{row} \\ \text{not new reference pixel}, & \text{otherwise} \end{cases} \tag{4}$$

The row scanning in one direction terminates until a couple of consecutive non-road pixels are found according to Eq.3. Thus, the row scanning procedure can obtain the left (or the right) boundary of the road region on the $n-$th row. Next, we should determine the first reference pixel on the $(n-1)$-th row. It is determined by two previous boundary pixels and the first reference pixel, i.e.

$$\text{Middle}_{n-1} = (\text{Left}_n + \text{Right}_n + \text{Middle}_n)/3 \tag{5}$$

where Middle_n is the position of the first reference pixel on the $n-$th row, Left_n and Right_n being the positions of two boundaries of the detected road region on the $n-$th row. After obtaining the first reference pixel on the $(n-1)$-th row, a new round of row scanning is performed as described above. The whole row scanning process will terminate when the row scanning in the top row is completed. An example of row scanning is shown in Fig. 3(b).

2) The Column Scanning: The detection result obtained by the row scanning is deficient, because the row scanning operates along the self-rotation direction of the LiDAR. If there exists a car in the middle of the road, the row scanning may neglect the road region beside one side of the car (as shown in Fig. 3(b)). To solve this problem, the column scanning is conducted in the LiDAR imagery. It scans along the column from the bottom to the top, determining whether the pixels from the location near LiDAR to the faraway region belong to the road region or not.

The column scanning process begins from the bottom row (the first reference point) to the top row in each column and ends when meeting the *non-road* points. Likewise, it also contains two steps: updating the reference pixel and determining the road boundary pixel. Unlike the row scanning, the first reference pixel in the column scanning is based on the row scanning result. Only the columns in which the pixels in several bottom rows belong to the road region are used in the column scanning. The first reference pixel for each chosen column is set to the one in the bottom row. Next, we scan its upper neighbor pixel in the same column. Eq.6 gives a criterion to determine whether the current pixel belongs to the road region or not,

$$P_C \begin{cases} \in \text{road region}, & \text{if } \|z_R - z_C\| > \Delta_{col} \\ \notin \text{road region}, & \text{otherwise} \end{cases} \tag{6}$$

where the height of the reference pixel P_R and the current pixel P_C is represented by z_R and z_C, respectively.

If the current pixel is located in the road region, by the criterion in Eq.7, we proceed to determine whether the current pixel should be updated to the new reference point or not,

$$P_C \begin{cases} \text{new reference pixel}, & \text{if } d_{xy}(P_R, P_C) \geq \delta_{col} \\ \text{not new reference pixel}, & \text{otherwise} \end{cases} \tag{7}$$

The column scanning in one column terminates until a couple of consecutive non-road points are found according to Eq.6. Thus, the column scanning procedure obtains the upper boundary of the road region. The whole column scanning process will terminate when all the chosen columns are processed. An example of column scanning is shown in Fig. 3(c).

3) Fusing the Row- and Column-Scanning: The row scanning may provide the wrong road boundary when an obstacle is located in the middle of the road, and the column scanning result is dependent on the row scanning result in the bottom rows. Therefore, we hope to combine the results of both the row- and column- scanning. The row scanning result is considered as the base result, while the column scanning result is used to fill the missing part near the obstacle for the row scanning. Fig. 3 shows the row scanning, the column scanning and the fusion results. It shows that the fusion result improves the detection performance.

Fig. 5: Left: the original Delaunay Triangulation result. Right: the improved Delaunay Triangulation result

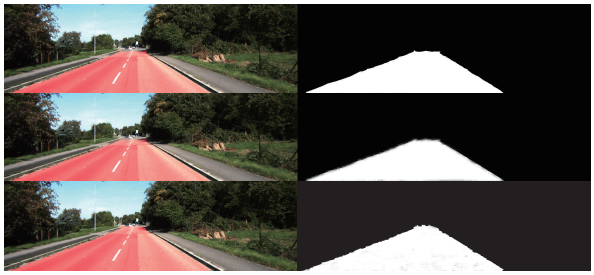

Fig. 6: The first row: the result and road mask by the Delaunay triangulation based method. The second row: the result and road mask by the Delaunay triangulation and guided image filtering based method. The third row: the result and road mask by the Total Generalized Variation based method.

D. Fusion of the LiDAR-Imagering Scanning with an Image-Guided Diffusion

After acquiring the road region in the LiDAR imagery, all the 3D points of that region can be projected into the image space based on the LiDAR-camera calibration parameters. But it makes the point cloud very sparse in the image space. Different from the works [1], [2] and [16], we regard the image as the guide information and conduct an upsampling operation to obtain a dense road region result in the image. In this section, we introduce two strategies: one is based on the Delaunay triangulation and guided image filter, and the other is based on the Total Generalized Variation (TGV).

1) Delaunay Triangulation (DT) and Guided Image Filter (GIF) Based Fusion: The Delaunay triangulation is a valid method to organize unordered points in a 2D plane. By doing this, points are associated with each other by triangles. We conduct a Delaunay triangulation on the LiDAR points of road region that have been projected to the image, as shown in Fig. 5. Note that some triangles near the road boundary have long edges, which may include the non-road region. As a result, these triangles with long edges will be omitted from the triangulation. Next, the remaining triangles are filled as the base road detection result in the image space.

However, the Delaunay triangulation neglects the texture information in the image, which may lead to a rough boundary for the road region. To fit the rough boundary into the smooth road boundary in the image, some edge-preserving filtering methods can be used. In this paper, we choose the guided image filter [17] to fit the boundary region, where the RGB image is regarded as the guided image and the rough road detection result (a binary image) is the filtered input image. The experimental result is also shown in Fig. 6.

2) Total Generalized Variation (TGV) Based Fusion: Although the Delaunay triangulation and guided image filter based fusion method could give promising results, there still exists a problem to solve: when a non-road region is included into the rough road detection result during the Delaunay triangulation, it cannot be removed from the road region result in the subsequent filtering process. In this section, we will use a TGV based method, which directly diffuses the sparse road region point into the gray-scale image space.

The work [18] is a classical TGV based method to achieve an image guided depth upsampling. It increases the resolution of the depth data from a low resolution sensor by considering the edge information from an image with high resolution and preserve the sharp edges based on the texture information at the same time. Moreover, it does not require a specific distribution in the image space for the depth data. Hence, an image guided diffusion problem can be transfered into a depth-data upsampling problem: the points in the road region and the ones in the non-road region are assigned to two different categories of depth data, The gray-scale image is treated as the guided image, and the final road detection result corresponds to the depth upsampling result.

Fig. 6 shows an experimental result. It is worth noting that the LiDAR points cannot cover the road region in the distant, which cannot be detected by the scanning based method. Nevertheless, this region can be diffused during the fusion process.

IV. EXPERIMENTS

A. Dataset

In this section, we will evaluate the performance of our proposed framework on the widely-used KITTI_ROAD dataset [19]. The KITTI_ROAD dataset consists of about 600 frames with synchronized stereo RGB images and LiDAR data. The LiDAR is Velodyne HDL-64E, which has 64 lasers to transmit and receive signals. Because the dataset directly provides the point cloud, we first re-project the point cloud back to the LiDAR imagery. In this paper, we set the angle resolution for each column of the LiDAR imagery to be 0.25°, so that the size of the original obtained LiDAR imagery is 64×1440. Due to a small HFOV of the camera and the existence of the vehicle's bonnet, we only choose the top 56 rows and the columns with the indexes from 540 to 900 to constitute the finally used LiDAR imagery.

The KITTI_ROAD dataset consists of three subsets: UU (urban unmarked road), UM (urban marked road) and UMM (urban multiple marked lanes). Each subset consists of about 100 frames for training and about 100 frames for testing. All the training frames provide the well-labeled ground-truth for validation, while the ground-truth of testing frames are not publicly available and one should upload the results to conduct the on-line validation. According to [19], the detected result is evaluated in the Bird Eye View (BEV) space. Six criteria are used to evaluate the accuracy, including Precision (PRE), Recall (REC), False Positive Rate (FPR), False Negative Rate (FNR), maximum F1-measure (MaxF) and Average Precision (AP).

TABLE I: Evaluation for the row scanning, the column scanning and the fusion of the row- and column- scanning on the KITTI training dataset.

Method	MaxF	AP	PRE	REC	FPR	FNR
row scanning	95.26%	88.20%	93.71%	96.85%	3.21%	3.15%
column scanning	92.14%	81.99%	86.88%	98.09%	7.31%	1.91%
fusion	95.34%	87.98%	93.47%	97.29%	3.35%	2.71%

TABLE II: Comparisons on the training dataset in BEV.

Dataset	Method	MaxF	AP	PRE	REC	FPR	FNR
UM_ROAD	S+DT	92.12%	85.01%	90.37%	93.95%	4.61%	6.05%
	S+DT+GIF	92.69%	**91.03%**	91.91%	93.49%	3.79%	6.51%
	S+TGV	**93.09%**	89.02%	**92.13%**	**94.07%**	**3.69%**	**5.93%**
UMM_ROAD	S+DT	95.55%	91.65%	95.60%	95.49%	4.80%	4.51%
	S+DT+GIF	95.83%	**94.76%**	95.84%	**95.83%**	4.54%	**4.17%**
	S+TGV	**96.05%**	93.91%	**96.45%**	95.65%	**3.84%**	4.35%
UU_ROAD	S+DT	90.46%	81.51%	87.03%	**94.18%**	4.99%	**5.82%**
	S+DT+GIF	90.79%	**88.47%**	**88.70%**	92.99%	**4.22%**	7.01%
	S+TGV	**91.08%**	85.2%4	88.51%	93.81%	4.33%	6.19%

B. Experiment Setting

The detailed parameter settings are listed as follows. For the flat region detection, $\alpha = 6$, $\beta = 0.8$ and $\gamma = 0.05$. For the row scanning, the height different threshold $\Delta_{row} = 0.02$ and the reference pixel threshold $\delta_{row} = 0.2$. For the column scanning, the height different threshold $\Delta_{col} = 0.08$ and the reference pixel threshold $\delta_{col} = 1$. The experiments were conducted on a standard PC with a RAM of 8G and a CPU of 3.5GHz. The algorithm was coded in single thread C++ under Windows 8. The average running time of row- and column- scanning is only 0.07 sec per frame, while DT, GIF and TGV spend much time. Note that our code is implemented without any optimization. The row and column scanning part is very suitable to parallel arithmetic, while the Triangulation, the guided image filter and the total generalized variation are suitable for GPU processing due to the large amount of matrix and tensor calculation.

C. Performance Evaluation

In this section, we first evaluate the performance of the scanning process, including the row scanning, the column scanning and the fusion of row- and column- scanning methods. Since the scanning is operated on the LiDAR imagery, the evaluation is also conducted based on 3D points. The result is showed in Table I. The MaxF measurement for the fusion method is larger than the ones of other two methods, which verifies the effectiveness of the fusion process.

Next, we will present both the qualitative and the quantitative evaluation results of our proposed approaches. Since the ground-truth of the testing frames are not available for the off-line evaluation, we first test our proposed methods in the training dataset. Table II show the evaluation results. To simplify the notation, the method using the row- and column- scanning and Delaunay triangulation is denoted as "S+DT", the method using the row- and column- scanning, Delaunay triangulation and guided image filtering is written as "S+DT+GIF", while the method using the row- and column- scanning and total generalized variation is denoted as "S+TGV". The MaxF value of "S+DT+GIF" is always larger than the one of "S+DT", which indicates that guided

TABLE III: Comparisons on the testing dataset in BEV.

Method	UM(MaxF)	UMM(MaxF)	UU(MaxF)	URBAN(MaxF)
S+TGV (Ours)	92.83%	95.72%	90.50%	93.56%
S+DT+GIF (Ours)	91.87%	95.56%	89.90%	93.00%
HID-LS[15]	92.03%	94.36%	89.10%	92.36%
LoDNN [12]	92.75%	96.05%	92.29%	94.07%
FTP [20]	91.20%	92.98%	89.62%	91.61%
HybridCRF [2]	90.99%	91.95%	88.53%	90.81%
NNP [21]	90.50%	91.34%	85.55%	89.68%
Up-Conv [4]	90.48%	93.89%	91.89%	92.39%
LidarHisto [11]	89.87%	93.32%	86.55%	90.67%
FusedCRF [1]	89.55%	89.51%	84.49%	88.25%

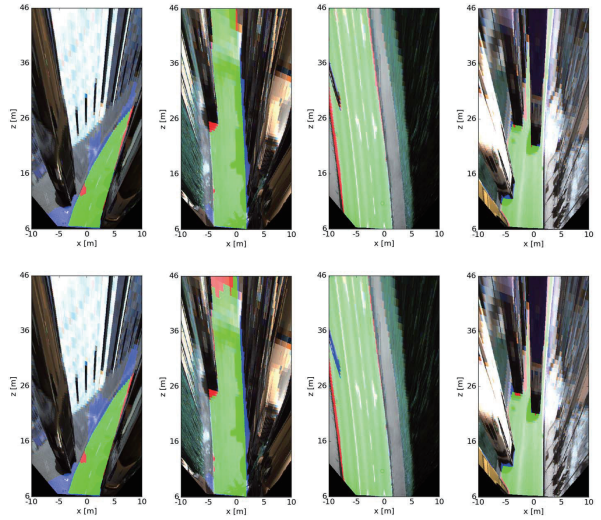

Fig. 7: First row: road detection results by our proposed method S+TGV in BEV. Second row: road detection results by our proposed method S+DT+GIF in BEV. The red areas are false negatives. The blue areas correspond to false positives. The green areas represent true positives.

image filtering is valid to fit the road boundary of the 3D points to the one in the image space. The MaxF value of "S+TGV" is largest, which indicates that total generalized variation based diffusion can fill the road area adequately.

We also evaluate our proposed methods on the testing dataset by the on-line evaluation. Table III shows the evaluation results. We compare our methods with some high-ranking methods, including LiDAR based methods ([12]), LiDAR-camera fusion based methods ([15], [2], [11], [1]), monocular based methods ([4], [20]), stereo based method ([21]), deep learning based methods ([4],[12]) and scanning based method ([15]). Both of our proposed two methods achieve the promising results. From the average results, the MaxF values of our proposed two methods are the highest except the deep learning based method [4], which extracts several complex features from the top-view image. Our proposed two methods are also better than the scanning based method [15], which verifies the effectiveness of the LiDAR imagery and the diffusion strategy. In particular, the MaxF measurements of our proposed methods are also lower than the one of [4] in the UU_ROAD dataset, because there does not exist drastic height changes around the road boundaries. Some detection results in BEV are showed in Fig.7.

V. Conclusion

In this paper, we propose a LiDAR-camera based road detection method. The point cloud is re-organized to generate a LiDAR imagery, in which a flat region is extracted as the rough candidate. Next, the row- and column- scanning strategy is conducted in the LiDAR imagery to detect a finer road region from the candidate. To fuse the point cloud with image information, we transform the point cloud to the image space based on the calibration parameters. At last, two image guided diffusing schemes are given to transform the sparse points to the dense road region in the image space. Experiments show that our method compares favorably with the state-of-the-art on the KITTI benchmark.

References

[1] L. Xiao, B. Dai, D. Liu, T. Hu, and T. Wu, "Crf based road detection with multi-sensor fusion," in *IV*, 2015.

[2] L. Xiao, R. Wang, B. Dai, Y. Fang, D. Liu, and T. Wu, "Hybrid conditional random field based camera-lidar fusion for road detection," *Information Sciences*, 2017.

[3] J. Fritsch, T. Kühnl, and F. Kummert, "Monocular road terrain detection by combining visual and spatial information," *TITS*, 2014.

[4] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep methods for monocular road segmentation," in *IROS*, 2016.

[5] M. Teichmann, M. Weber, J. M. Zöllner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," *arXiv:1612.07695*.

[6] C. C. T. Mendes, V. Frémont, and D. F. Wolf, "Exploiting fully convolutional neural networks for fast road detection," in *ICRA*, 2016.

[7] R. Mohan, "Deep deconvolutional networks for scene parsing," *arXiv:1411.4101*.

[8] A. Asvadi, C. Premebida, P. Peixoto, and U. Nunes, "3d lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes," *Robotics and Autonomous Systems*, 2016.

[9] X. Hu, F. S. A. Rodriguez, and A. Gepperth, "A multi-modal system for road detection and segmentation," in *IV*, 2014.

[10] T. Chen, B. Dai, R. Wang, and D. Liu, "Gaussian-process-based real-time ground segmentation for autonomous land vehicles," *Journal of Intelligent & Robotic Systems*, 2014.

[11] L. Chen, J. Yang, and H. Kong, "Lidar-histogram for fast road and obstacle detection," in *ICRA*, 2017.

[12] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast lidar-based road detection using convolutional neural networks," *arXiv:1703.03613*.

[13] P. Y. Shinzato, D. F. Wolf, and C. Stiller, "Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion," in *IV*, 2014.

[14] Z. Liu, S. Yu, X. Wang, and N. Zheng, "Detecting drivable area for self-driving cars: An unsupervised approach," *arXiv:1705.00451*.

[15] S. Gu, Y. Zhang, J. Yang, and H. Kong, "Lidar-based urban road detection by histograms of normalized inverse depths and line scanning," in *ECMR*, 2017.

[16] Z. Liu, S. Yu, X. Wang, and N. Zheng, "Detecting drivable area for self-driving cars: An unsupervised approach," *arXiv1705.00451*.

[17] K. He, J. Sun, and X. Tang, "Guided image filtering," *TPAMI*, 2013.

[18] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *ICCV*, 2013.

[19] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *ITSC*, 2013.

[20] A. Laddha, M. K. Kocamaz, L. E. Navarro-Serment, and M. Hebert, "Map-supervised road detection," in *IV*, 2016.

[21] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *NIPS*, 2015.