Multi-modal curb detection and filtering

Sandipan Das[1,2], Navid Mahabadi[2], Saikat Chatterjee[1], Maurice Fallon[3]

***Abstract*— Reliable knowledge of road boundaries is critical for autonomous vehicles navigation. We propose a robust curb detection and filtering technique based on the fusion of camera semantics and dense lidar point clouds. The lidar point clouds are collected by fusing multiple lidars for robust feature detection. The camera semantics are based on a modified EfficientNet architecture which is trained with labeled data collected from onboard fisheye cameras. The point clouds are associated with the closest curb segment with L_2-norm analysis after projecting into the image space with the fisheye model projection. Next, the selected points are clustered using unsupervised density-based spatial clustering to detect different curb regions. As new curb points are detected in consecutive frames they are associated with the existing curb clusters using temporal reachability constraints. If no reachability constraints are found a new curb cluster is formed from these new points. This ensures we can detect multiple curbs present in road segments consisting of multiple lanes if they are in the sensors' field of view. Finally, Delaunay filtering is applied for outlier removal and its performance is compared to traditional RANSAC-based filtering. An objective evaluation of the proposed solution is done using a high-definition map containing ground truth curb points obtained from a commercial map supplier. The proposed system has proven capable of detecting curbs of any orientation in complex urban road scenarios comprising straight roads, curved roads, and intersections with traffic isles.**

I. Introduction

Robust environmental perception is a fundamental aspect of autonomous driving that is important for road safety and efficiency and contributes to technical problems such as path planning, control, and localization. Highly dynamic driving environments can pose critical safety challenges for self-driving vehicles. Objects (stationary or moving), as well as road construction, can change the geometry of the road and result in degradation to localization and planning. Curbs define the road boundary and provide useful information for vehicle navigation; as a result, accurately detecting and tracking them is important.

Over the past few years, there have been numerous methods proposed to detect and extract curb features using either a single sensor or a combination of sensor modalities. Most curb detection systems use lidar and cameras [1]. Lidar sensors have been frequently used to detect curb features as the curbs are inherently geometric features [2], [3], [4], [5]. Vision based processing techniques have also been proposed including [6], [7], [8]. Because lidar and vision have different failure modes, sensor fusion has become popular in recent years which exploits the best properties of both the sensors – camera images for semantics and lidar for depth information[9], [10], [11].

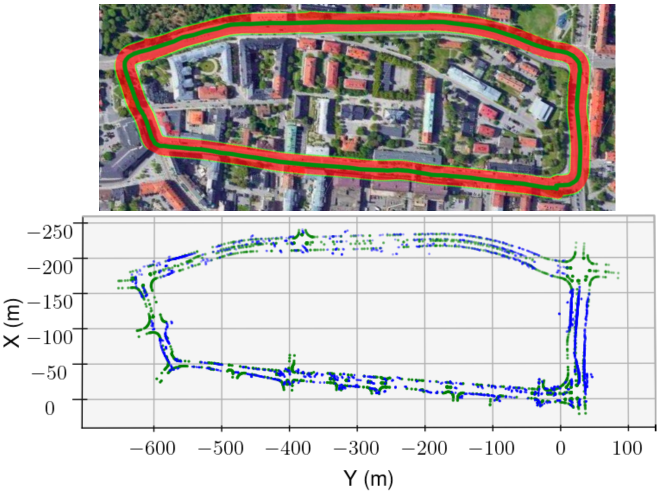


Fig. 1: *Top:* The route of the data collection vehicle. *Bottom:* Detected curb features using our proposed method (blue points) and the ground truth curb features (green points) from a commercial map supplier.

A. Motivation

Temporary change to an environment may occur due to various factors. Hence, it is important to incorporate those updates in the mapping module so that the planner and control modules may react accordingly. Consequently, we need an inhouse methodology from which we can extract the necessary features of an environment into our mapping module. In addition, we would have the ability to create curb maps in restricted areas to which commercial map suppliers do not have access.

B. Contribution

Our work falls under the category of association of camera semantics with lidar depth. We show the results of our proposed method in Fig. 1 and to achieve that our specific contributions are:

- Detection and association of multiple curbs with unsupervised DBSCAN clustering.
- Outlier removal using Delaunay-based filtering method which needs less parametric tuning than RANSAC-based polynomial fitting regression constraints.

II. Related Work

Aspects of curb detection have been studied extensively in the context of autonomous driving. Multiple solutions have been proposed that use camera, lidar, or fusion of both.

[1] KTH EECS, Sweden. {sandipan,sach}@kth.se
[2] Scania, Sweden. {sandipan.das,navid.mahabadi}@scania.com
[3] Oxford Robotics Institute, UK. mfallon@robots.ox.ac.uk

In the early 2000s, traditional computer vision methods were proposed for curb detection. In [12], the authors proposed a histogram filter with a threshold to locate curb points. Whereas, in [2], a Kalman filter based tracking of the threshold-based detected curb points was proposed. By the mid-2000s, Hough transform based method [13] on 2D lidar data was explored under the assumption that the terrain was flat. Polyfit on a digital elevation map (DEM) created from the stereo vision was explored in [6], [7], while the authors in [14] used height images from lidars to create DEM and applied polynomial fitting. In [8] the authors proposed to create a DEM from ego-motion and filtering for curb points. As 3D lidar was adopted by researchers after the mid-2000s, a lot of the work for curb detection was based on ground plane segmentation on 3D data [15], [4], [5].

In recent years, with better network architectures for semantic segmentation, works began to fuse camera semantics with lidar depth [9], [10]. Instead of relying on geometric attributes of curbs, the semantic information identified specific curb pixels. Then the depth was obtained from the lidar projected point clouds in the image plane. A recent work, CurbScan [11], proposed to fuse an additional ultrasonic sensor for lateral distance information with curb tracking using a Kalman filter.

In contrast to prior works, we propose an agglomerative unsupervised clustering to detect multiple curbs in unfiltered point clouds from re-projected camera semantics. We also apply a Delaunay-based filter to remove outliers among the detected curb points. Finally, we transform all the detected curbs into the global frame using GNSS (Global Navigation Satellite System) and check their consistency quantitatively with respect to the ground truth (GT) curb points obtained from a prior hand annotated HD curb map.

III. Methodology

Prior calibration of the sensors is a fundamental prerequisite for sensor fusion. Additionally, an important feature of our fusion technique for feature association is that the lidar point clouds are motion corrected and transformed so as to be equivalent to that recorded at the time stamp of an available camera frame using the approach introduced in [16]. This helps in data from all the sensors need to be properly time synchronized. In the following sub-sections we outline the procedure we adapted for our approach.

A. Sensor setup and reference frames

The data collection vehicle consisted of two lidars and two cameras. The reference frames and the field of view (FoV) of the sensors are shown in Fig. 2. The vehicle base frame $\mathtt{B}$ is located on the center of the rear-axle of the vehicle. Sensor readings from lidars and cameras are represented in base frame $\mathtt{B}$, as ${}_{\mathtt{B}}\mathtt{L}_k$ and ${}_{\mathtt{B}}\mathtt{C}_k$ respectively, where k represents the location of the sensor in the vehicle. For example, $\mathtt{L}_L$ and $\mathtt{L}_R$ represent the front left and front right lidar frames. The pose information from the GNSS is reported in the UTM frame, $\mathtt{W}$ (World frame). The IMU frame, $\mathtt{I}$, of the GNSS is shown in Fig. 2. All the inter-sensor transformations are carried out using prior calibration parameters.

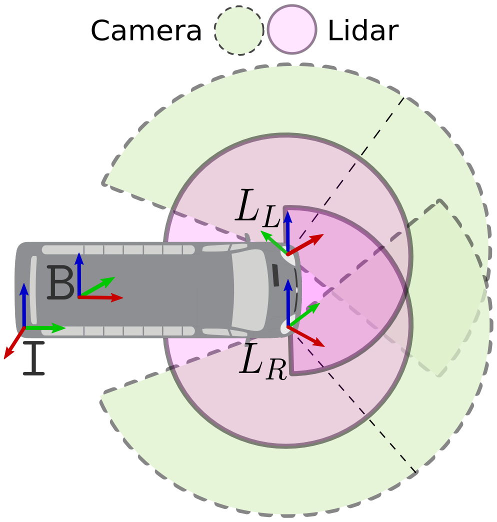


Fig. 2: Illustration of the FoV of two front lidars and four cameras positioned around the data collection vehicle. However, only the two front facing cameras were used for our experiment. The vehicle base frame $\mathtt{B}$ is located at the center of the rear axle. The sensor frames of the lidars are $\mathtt{L}_L$ and $\mathtt{L}_R$, representing front-left and front-right lidars respectively. The IMU frame of the GNSS is represented as $\mathtt{I}$.

B. Semantic association with lidar points

The semantic features from lidar were extracted by associating the projected point clouds with the segmented camera images. The image segmentation was performed using a trained Efficient-Net model.

1) Efficient-Net for semantic segmentation: Since AlexNet [17] won the 2012 ImageNet competition, Convolutional Neural Networks (ConvNets) have been increasingly used as the de facto standard for image segmentation tasks. Although higher accuracy is critical for autonomous driving applications, we have already hit the hardware memory limit for the image segmentation tasks using ConvNets. Thus further accuracy gains will require better efficiency.

The authors of EfficientNet [18] illustrated that model scaling can be achieved by carefully balancing network depth, width, and resolution, leading to a better performance with the fixed amount of computation resources. Based on this study, we used a modified architecture of EfficientNet for semantic segmentation. The network architecture backbone is shown in TABLE I, where each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. MBConv layers represent mobile inverted bottlenecks from the MobileNetV2 architecture [19], where squeeze-and-excitation optimization [20] has also been added on top of it. To upsample the network output to its original input resolution a bilinear interpolation was used in the decoder architecture.

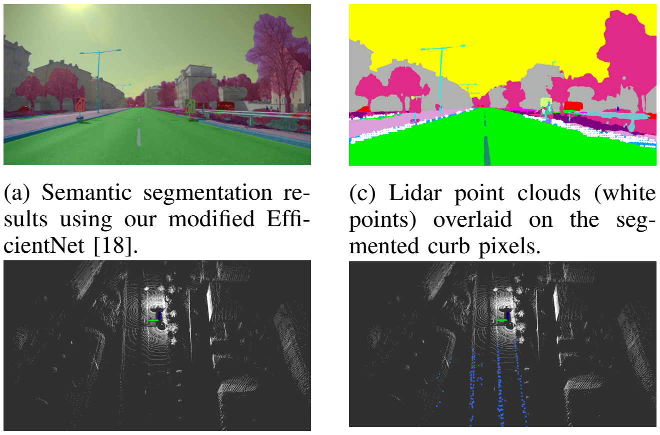
To achieve robustness we trained the EfficientNet model on a diversified set of scenarios (city driving, snowy conditions, dessert area, suburban area). The overall training accuracy

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	640 × 1024	32	1
2	MBConv1, k3x3	320 × 512	16	1
3	MBConv6, k3x3	160 × 256	24	2
4	MBConv6, k5x5	80 × 128	40	2
5	MBConv6, k3x3	40 × 64	80	3
6	MBConv6, k5x5	40 × 64	112	3
7	MBConv6, k5x5	20 × 32	192	4
8	MBConv6, k3x3	20 × 32	320	1
9	Segmentation Head	640 × 1024	24	-

TABLE I: EfficientNet-B0 baseline network architecture.

is 79.10% and the mean IoU (Intersection over Union) is 0.495. For the curb class the accuracy is 67.20% and the IoU is 0.557. The results of the segmentation on a sample frame can be visualized in Fig. 3a.

2) Association of curb semantics with lidar depth: The cameras mounted on our platform have fisheye lenses. Hence, we extracted the curb points by doing a fish-eye projection [21] of the fused lidar points in image space and chose points closer to the curb pixels within a bound of ±3 pixels. Finally, the curb points are re-projected back to the base frame B. The results of curb extraction from semantic association can be visualized in Fig. 3d.



(a) Semantic segmentation results using our modified EfficientNet [18].

(c) Lidar point clouds (white points) overlaid on the segmented curb pixels.

(b) Fused lidar point clouds from lidar sensors.

(d) Curb semantics (blue points) with the fused point cloud.

Fig. 3: Semantic association with lidar point cloud.

C. Unsupervised clustering and filtering

The point cloud association technique may give us noisy points due to various reasons such as synchronization quality of the logs, calibration parameters, or camera projection model. To remove outliers we can apply filtering based on the geometric structures of the curbs. However, since we do not know the pre-defined number of curbs in advance, it makes it difficult to apply a polynomial fitting on the extracted curb points. To overcome this problem we first find a set of unsupervised clusters and associate newly detected curb points to the relevant clusters based on spatial density.

1) Iterative cluster association: We iteratively chose the extracted curb point clouds and applied unsupervised spatial clustering. The boundary points of a cluster are the points which are furthest from the cluster centroids. If the L_2-norm of the boundary points in the new clusters were less than a pre-defined threshold from the old cluster boundary points we merged the clusters. This operation helped in identifying the number of curb segments. The result of our clustering with DBSCAN [22] is shown in Fig. 4.

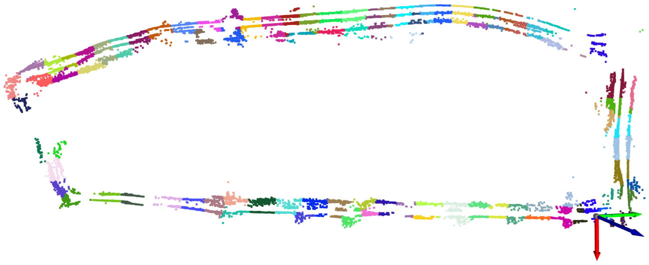


Fig. 4: Iterative feature point clustering using DBSCAN [22]. Random colors indicate different clusters detected.

2) Delaunay filtering: Let $\mathcal{S}$ be a set of points in $\mathbb{R}^n$ with distance Euclidean function d. The Voronoi diagram [23] is the partition of the space into Voronoi regions, R, such that

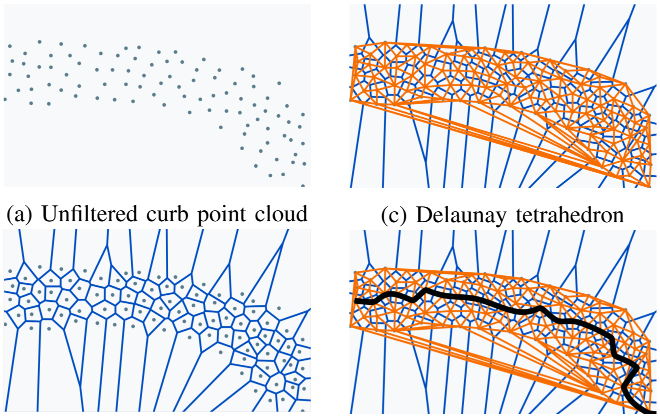
$$R_k = \{s \in \mathcal{S} \mid d(s, R_k) \leq d(s, R_j) \; \forall \; j \neq k\}. \tag{1}$$

Voronoi graphs have been used in motion planning algorithms for obstacle avoidance [24], [25]. But in the context of filtering of curb points, this is a novel approach. Delaunay triangulation is the dual of the Voronoi diagram. Delaunay triangulation [26] on $\mathcal{S}$, DT($\mathcal{S}$) is a triangulation such that no point in $\mathcal{S}$ is inside the circum-hypersphere of any n-simplex in DT($\mathcal{S}$). For $\mathbb{R}^3$ it is called Delaunay tetrahedralization. The Voronoi vertex corresponding to a Delaunay tetrahedron is the center of the circumscribing sphere of the tetrahedron. Let $[x \quad y \quad z]$ be the center and the four points of the tetrahedron be $[x_i \quad y_i \quad z_i], \forall i = 1(1)4$. Then the center is found by solving the following equation:

$$\begin{aligned} &(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = \\ &(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2, \forall j = 2, 3, 4 \\ \implies &2(x_j - x_1)x + 2(y_j - y_1)y + 2(z_j - z_1)y = \\ &(x_j^2 + y_j^2 + z_j^2) - (x_1^2 + y_1^2 + z_1^2), \forall j = 2, 3, 4 \end{aligned} \tag{2}$$

Voronoi sub-graph of the Delaunay tetrahedra is computed by filtering large radii of circumscribing spheres from the centers computed. This removes tetrahedra outside the point volume and removes the outliers. The shortest Euclidean path in the Voronoi sub-graph connecting the start and the end points give us the medial axis. The points closer to the medial axis gives us the filtered point cloud corresponding to the curbs. An illustration of the process is shown in Fig. 5.

3) RANSAC filtering: RANSAC [27] is an iterative algorithm for the robust estimation of parameters from a subset of inliers from the complete data set. For fitting the curb points to an estimator we used a third-order polynomial. However, an automatic parameter tuning to estimate the degree of the polynomial was performed for each curb segment to find out the best set of inliers.



(a) Unfiltered curb point cloud

(c) Delaunay tetrahedron

(b) Voronoi regions

(d) Medial axis

Fig. 5: Curb inlier selection with Delaunay filtering.

IV. Experimental Results

A. Dataset

We collected the data from a Scania autonomous bus platform mounted with two lidars and two front facing cameras for a route length of ≈1.5 Kms. The ground truth (GT) curb features are provided by a map supplier. All the sensing data was synchronized using PTP (Precision Time Protocol) synchronization and converted to rosbags for evaluation. We evaluated the generated curb points by manually selecting the corresponding GT curb points. The point cloud selection tool for association was developed using open3d [28]. Since manual association is a tedious process we also propose a mechanism for automatic evaluation of the clusters. The clustering algorithms are evaluated offline from scikit-learn [29] package.

B. Manual segment-wise association and evaluation

We associate the GT points from the map supplier segment-wise. For evaluation, we fit a polynomial to the GT points. Then we sample points from the polynomial and associate them to the Delaunay filtered points and RANSAC filtered points as shown in Fig. 6. We compute the normalized L_2-norm (based on the number of points selected) for evaluation metrics.

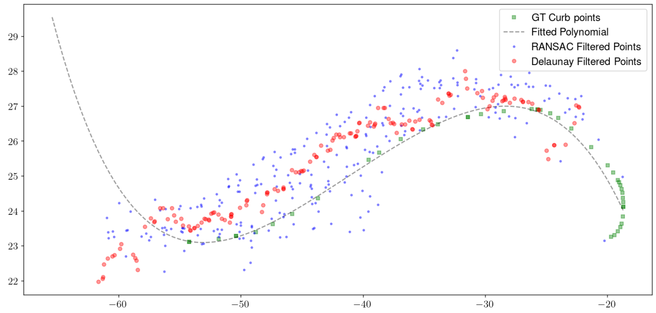


Fig. 6: The curb points are extracted by applying RANSAC (blue points) and Delaunay filtering (red points). A polynomial fitting is performed (black line) to the GT points (green) after which the L_2-norm between the GT polyline and the filtered points are calculated.

C. Automatic segment-wise association and evaluation

To automatically evaluate the generated curb points we measure the Chamfer distance (CD) [30] of each cluster segment. The CD between two set of point clouds $P_1, P_2 \subseteq \mathbb{R}^3$ is defined as:

$$\mathrm{CD}\left(P_1, P_2\right) = \frac{1}{|P_1|}\sum_{a\in P_1}\min_{b\in P_2}\|a-b\|_2^2 + \frac{1}{|P_2|}\sum_{b\in P_2}\min_{a\in P_1}\|b-a\|_2^2 \tag{3}$$

We also evaluate the effect of different unsupervised clustering algorithms including Agglomerative Clustering [31], BIRCH [32], DBSCAN [22] and OPTICS [33] for our evaluation metrics. The CD and the number of detected filtered curb points for our evaluation are reported in TABLE II.

We observe that Delaunay filtering identifies inliers that more closely correspond to the GT points than a generic RANSAC-based filtering both for manual and automatic segment wise association as the L_2-norm and CD are lower for Delaunay filtering compared to RANSAC. We also observe that the Delaunay filtering selects lesser inlier points compared to RANSAC. However, the inliers correspond to the GT points more closely as shown in Fig. 6. We conclude that density based unsupervised algorithm for segment clustering works best for fitting an arbitrary number of curbs based on the computed CD. The result of our proposed method showing the final selected curb points is shown in Fig. 1.

Manual segment-wise association		
No Clustering	**Normalized L_2-Norm**	**# Detected Points**
RANSAC Filtering	27.659	9578
Delaunay Filtering	**19.947**	6904
Automatic segment-wise association		
Outlier Removal (RANSAC)	**Chamfer Distance**	**# Detected Points**
Agglomerative Clustering	17.427	3489
BIRCH	19.596	1351
DBSCAN	**17.220**	5314
OPTICS	18.370	7446
Outlier Removal (Delaunay)	**Chamfer Distance**	**# Detected Points**
Agglomerative Clustering	15.418	3924
BIRCH	16.165	3492
DBSCAN	**14.753**	6678
OPTICS	15.870	4415

TABLE II: Evaluation Metrics

V. Conclusion

We proposed a multi-modal curb detection and mapping algorithm with a novel filtering approach using 3D-Delaunay tetrahedra. We demonstrated the detection of arbitrary number of curbs with our clustering approach. Our evaluation indicates that Delaunay filtering outperforms traditional RANSAC based filtering approach for curb outlier removal.

To extend this work further, we would study the instability of the medial axis generation under different noise conditions. We would also like to extend the semantic association to other infrastructure features like road lines, traffic lights, pedestrian paths. We also strive to benchmark our solution on open source datasets by retraining our semantic segmentation model on un-distorted images.

References

[1] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, Apr. 2014.

[2] W. Wijesoma, K. Kodagoda, and A. Balasuriya, "Road-boundary detection and tracking using ladar sensing," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 456–464, 2004.

[3] G. Wang, J. Wu, R. He, and S. Yang, "A point cloud-based robust road curb detection and tracking method," *IEEE Access*, vol. 7, pp. 24 611–24 625, 2019.

[4] C. Fernández, R. Izquierdo, D. F. Llorca, and M. Sotelo, "Road curb and lanes detection for autonomous driving on urban scenarios," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 1964–1969.

[5] T. Chen, B. Dai, D. Liu, J. Song, and Z. Liu, "Velodyne-based curb detection up to 50 meters away," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 241–248.

[6] F. Oniga and S. Nedevschi, "Polynomial curb detection based on dense stereovision for driving assistance," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1110–1115.

[7] J. Siegemund, D. Pfeiffer, U. Franke, and W. Förstner, "Curb reconstruction using conditional random fields," in *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 2010, pp. 203–210.

[8] M. Kellner, M. E. Bouzouraa, and U. Hofmann, "Road curb detection based on different elevation mapping techniques," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 1217–1224.

[9] S. E. C. Goga and S. Nedevschi, "Fusing semantic labeled camera images and 3d lidar data for the detection of urban curbs," in *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2018, pp. 301–308.

[10] S. E. Catalina Deac, I. Giosan, and S. Nedevschi, "Curb detection in urban traffic scenarios using lidars point cloud and semantically segmented color images," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3433–3440.

[11] I. Baek, T.-C. Tai, M. M. Bhat, K. Ellango, T. Shah, K. Fuseini, and R. R. Rajkumar, "Curbscan: Curb detection and tracking using multi-sensor fusion," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–8.

[12] R. Aufrere, C. Mertz, and C. Thorpe, "Multiple sensor fusion for detecting location of curbs, walls, and barriers," in *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683)*. IEEE, 2003, pp. 126–131.

[13] S.-H. Kim, C.-W. Roh, S.-C. Kang, and M.-Y. Park, "Outdoor navigation of a mobile robot using differential gps and curb detection," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 3414–3419.

[14] J. Stuckler, H. Schulz, and S. Behnke, "In-lane localization in road networks using curbs detected in omnidirectional height images," *VDIBERICHT*, vol. 2012, p. 151, 2008.

[15] S. El-Halawany, A. Moussa, D. D. Lichti, and N. El-Sheimy, "Detection of road curb from mobile terrestrial laser scanner point cloud," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, no. 5/W12, 2011.

[16] D. Wisth, M. Camurri, S. Das, and M. Fallon, "Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1004–1011, 2021.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NIPS*, vol. 25, 2012.

[18] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[20] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[21] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[22] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[23] J. W. Brandt and V. R. Algazi, "Continuous skeleton computation by voronoi diagram," *CVGIP: Image understanding*, vol. 55, no. 3, pp. 329–338, 1992.

[24] O. Takahashi and R. Schilling, "Motion planning in a plane using generalized voronoi diagrams," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 143–150, 1989.

[25] M. Garber and M. C. Lin, "Constraint-based motion planning using voronoi diagrams," in *Algorithmic Foundations of Robotics V*. Springer, 2004, pp. 541–558.

[26] S. Fortune, "Voronoi diagrams and delaunay triangulations," *Computing in Euclidean geometry*, pp. 225–265, 1995.

[27] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, jun 1981.

[28] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[30] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," Tech. Rep., 1977.

[31] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 407–416.

[32] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," *ACM sigmod record*, vol. 25, no. 2, pp. 103–114, 1996.

[33] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.