# Road Curb Detection: ADAS for a Road Sweeper Vehicle*

Ivan Bilić, Goran Popović, Tibor Bataljak Savić,
Ivan Marković, and Ivan Petrović[1]

[1]University of Zagreb Faculty of Electrical Engineering and Computing
Laboratory for Autonomous Systems and Mobile Robotics
name.surname@fer.hr

**Abstract.** Recent development of the concept of smart cities has led to an increasing demand for advanced technological solutions that drive forward the design and capabilities of utility vehicles operating in urban environments. One possibility is to exploit recent advances in computer vision to introduce a certain level of autonomy into some of the vehicle's functionalities, e.g., an advanced driver-assistance system. Modern road sweeper vehicles are designed to possess multiple systems for maintaining the road quality and city cleanliness, such as brushes, vacuums, and great vehicle maneuverability. Introducing autonomy to these control systems lowers the burden on the human operator, thereby increasing work efficiency and overall safety, as well as making a positive impact on worker health. This paper considers a 3D curb detection system that supports autonomous road sweeping. In order to achieve this, we utilize a vision-based approach that leverages stereo depth estimation and a pre-trained semantic segmentation model. In addition, we implement a simple LiDAR-based curb detection baseline. Finally, we collected our own dataset comprised of driving sequences resembling our use-case, which is used to conduct qualitative experiments.

**Keywords:** curb detection · driver assistance system · utility vehicles.

## 1 Introduction

The global impacts of climate change, increasing air pollution and population in urban areas, and the development of the concept of smart cities, which represents the key strategic backbone for the development of cities in the 21st century, are making an impact on the industry of utility vehicles. The need for technology that can help reduce adverse effects (e.g., the impact of air pollution on

* This work has been supported by the European Regional Development Fund under the grant KK.01.2.1.02.0115 – Development of environmentally friendly vehicle for cleaning public surfaces with autonomous control system based on artificial intelligence (EKO-KOMVOZ).

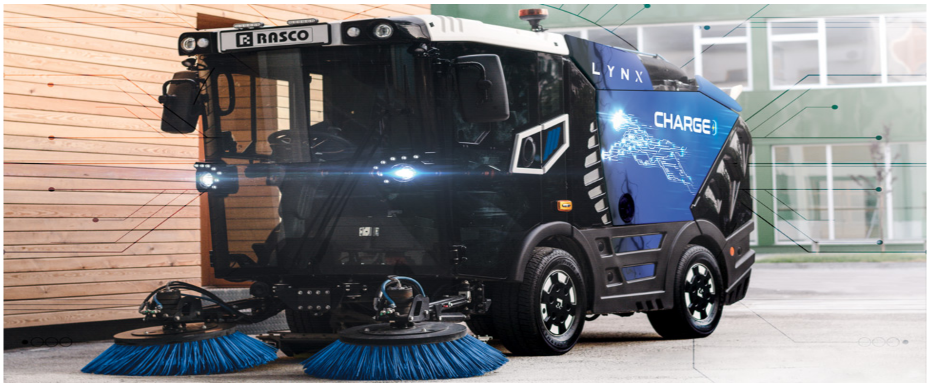


Fig. 1: RASCO LYNX Charge - an electric road sweeping utility vehicle.

human health), as well as increase efficiency and safety is becoming more and more emphasized. Utility vehicles, such as road sweepers, can be designed to possess multiple systems for road quality maintenance and city cleanliness, such as brushes and suction mouth (Fig. 1), in order to keep the roads and streets clean and free of undesirable objects. Furthermore, brush and suction mouth control pose additional load on the human operator, forcing them to perform multiple tasks simultaneously.

Presuming that the vehicle is always on the road, a reliable curb detection system acts as an enabler for autonomous cleaning, thereby lowering the load on the human operator. Hence, introducing autonomy in brush and suction mouth control systems can have a positive impact on both the efficiency and health of the human operator, while also contributing to the overall safety of the vehicle and its environment. Existing work on road curb detection is mostly comprised of 3D methods that obtain 3D information using sensors such as light detection and ranging (LiDAR) [10][4] or rely on stereo vision only [7][1]. Furthermore, in [3] authors investigate fusing semantic information with 3D data. Finally, real-time semantic segmentation for autonomous driving applications is a well active area of research [9][2].

In this paper, we investigate the problem of implementing a real-time system for road curb detection with the aim of supporting autonomous brush and suction mouth control. To tackle this problem, we first design a LiDAR-camera sensor rig that we mounted on the utility vehicle shown in Fig. 1 and used it to collect the dataset. This processes is described in Section 2. Furthermore, we consider two different, simple approaches for curb detection, discussed in Section 3; a vision-based, and a LiDAR-based approach, respectively. Section 4 describes our calibration procedure, while the conclusion is given in Section 5.

## 2 Dataset collected with the Lynx utility vehicle

The first task was to collect a dataset comprised of driving sequences in an urban environment, collected by driving the road sweeping utility vehicle shown in Fig. 1 throughout the streets of Zagreb. Concretely, the data was collected through seven driving sequences (∼15 minutes each), each on a different day, during daytime and under normal weather conditions, i.e., in the absence of fog, rain, and snow. Prior to data collection process, we mounted the custom rig system on the vehicle that consisted of two color cameras and a solid-state LiDAR. The rig system was fixed on the vehicle's windshield from the inside, using two vacuum plates, as shown in Fig. 2. Images were captured using two FLIR's Blackfly S GigE color cameras. These are global shutter cameras that offer high image resolution of $2048 \times 1536$, as well as high frame rate of 35 frames per second. As for the LiDAR, we used RS-LiDAR-M1, an automotive grade solid-state LiDAR designed for mass production vehicles. This LiDAR has a long range (up to 200m), with horizontal and vertical field-of-view of 120° and 25°, respectively. It is also characterized by a decent horizontal resolution of 0.2°, in addition to operating on a satisfying rate of 10 Hz. Finally, the calibration process, i.e., the process of obtaining the camera intrinsic and extrinsic parameters with respect to each other and the LiDAR, was carried out before and after each driving sequence, to check for considerable parameter changes.

The cameras were powered through a Power-Over-Ethernet (PoE) switch, while the LiDAR has it's own power supply. However, it was also connected to the same switch, thus cameras' and LiDAR measurements were synchronized via Generalized Precision Time Protocol (gPTP). As for the device drivers, both LiDAR and camera device drivers are publicly available and provided by the manufacturers. LiDAR drivers support Robot Operating System (ROS). However, this is not the case for FLIR cameras. Since we used ROS to implement the driver assistance software, we modified the camera drivers accordingly and wrapped them as a ROS node[1].

## 3 Road curb detection

The goal of this section is to describe the pipeline that we engineered in order to estimate the road curb in 3D space. To this end, we implemented two road curb detection methods. First, we describe a vision-based method for curb detection, i.e., one without exploiting LiDAR data and then we describe a LiDAR-based method which operates on the point cloud data only. All the processing components are implemented as ROS nodes, using Python and C++.

### 3.1 Vision-based curb detection

By vision-based curb detection, we refer to estimating the 3D points belonging to the road curb using only color images. To this end, we developed a pipeline

[1] https://bitbucket.org/unizg-fer-lamor/stereo_camera_driver

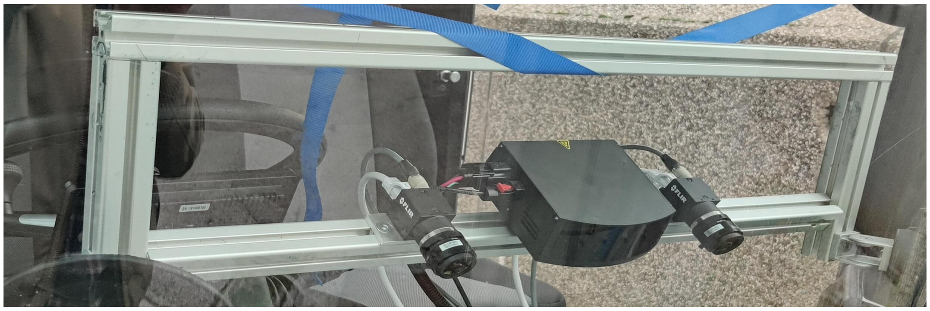


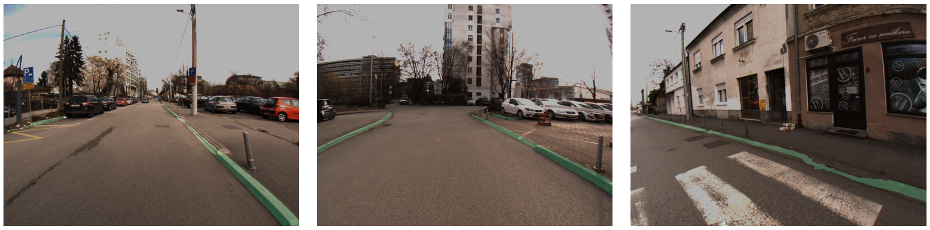
Fig. 2: A custom sensor rig system used to collect the data. Two color cameras and a solid-state LiDAR are fixed on the rig. The rig is attached to the windshield of the utility vehicle using two vacuum plates, and an extra band for safety.

that consists of stereo depth estimation, and a semantic image segmentation model, with “curb” in its class definition. Once the depth and semantic maps are obtained, a set of 3D points belonging to the curb can easily be found.

To recover the depth maps, we use Semi-Global-Matching (SGM) [5] – a stereo depth estimation algorithm that combines the concepts of global and local stereo methods for pixel-wise matching. The algorithm takes rectified stereo image pairs and outputs a disparity map. While different SGM variants have been proposed [6], we use the publicly available OpenCV’s implementation. A good trade-off between accuracy and computational cost makes SGM a favorable choice for many practical applications [6].

In order to identify the curb in an RGB image, we use semantic segmentation which outputs pixel-level labels belonging to a predefined set of classes. However, a publicly available dataset that resembles our use-case containing pixel-wise or 3D curb annotations was not available. Most datasets contain “road” and “sidewalk” classes, so the curb could be approximated as a boundary between those two classes. Nonetheless, we opted for a pre-trained model that explicitly contains the curb in its class definition. To this end, we use a road segmentation model[2] developed by Intel, trained to segment each pixel into one of the four classes; road, curb, mark, and background. Unfortunately, only the pre-trained model’s binary is publicly available – source code and training data are not. Nevertheless, the model is evaluated on 500 images of Mighty AI dataset, achieving 0.727 mIOU and 83.1% accuracy on the “curb” class with high computational efficiency, making it adequate for our use-case. We conducted qualitative experiments on our data to further test the model’s performance, using Intel’s Neural Compute Stick 2 (NCS2) for model inference.
Examples of qualitative inspection are depicted in Fig. 3. The model generalizes well on our dataset, successfully detecting the road edge for different types of curbs (Fig. 3a, Fig. 3b), and even drains (Fig. 3c), although less consistently. On the contrary, we expect degraded performance for types of curbs that are

[2] https://docs.openvino.ai



(a) (b) (c)

Fig. 3: Examples of road curb segmentation on our custom dataset. The curb mask is overlaid on the images and shown in green. Qualitative inspection reveals satisfactory model generalization capabilities.

under-represented in the training data, as well as in poor visibility and lighting conditions.

In general, given a 3D point $(X, Y, Z)^T$ in the camera coordinate frame, its projection on the image plane is governed by:

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{1}$$

where $(u, v, 1)^T$ denote the pixel coordinates, $\lambda$ is the scale factor, and $K$ represents the camera intrinsics, obtained through the camera calibration process. Thereby, the back-projection, i.e., the inverse operation is given by:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \lambda \mathbf{K}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} . \tag{2}$$

The curb pixel coordinates $(u, v)^T$ are provided by the semantic mask, while the depth map contains the $Z$-coordinates. Thus, given the semantic mask and (2), we obtain a set of 3D points corresponding to the curb.

### 3.2 LiDAR-based curb detection

In addition to vision-based curb detection, we also employ a simple LiDAR-based curb detection method. Our implementation is inspired by [4], where a road edge localization method is proposed and considered in the context of driver-assistance systems, among others. The key idea is that a set of 3D points belonging to a curb possesses certain geometric features, based on which the curb can be differentiated from other parts of the scene. That is, LiDAR data is processed to compute a geometric feature in the form of the angular distance to the ground normal. The remainder of the chapter describes the developed implementation.

Given that our application is an urban road driving scenario, we assume that the vehicle is always on the road. Consequently, extraction of the ground plane is a must. We first fit the plane from the point cloud using Random sample consensus (RANSAC), considering only points farther than 2m and closer than 20m from the LiDAR. We use 300 iterations and set the distance threshold for inlier selection to 0.05m. When RANSAC converges, parameters of the best model are used together with the corresponding inlier set.

Prior to calculating the geometric features, the point cloud is transformed in the ground plane's coordinate frame, given that it's defined in the LiDAR coordinate frame. This transformation is computed by using the ground plane parameters estimated with RANSAC. First, the magnitude of the ground plane normal is computed. The ground plane normal $\mathbf{z}$ is defined as $(a, b, c)^T$, where $a$, $b$ and $c$ are the ground plane parameters and the magnitude $\|\mathbf{z}\|$ is defined as $\|\mathbf{z}\| = \sqrt{a^2 + b^2 + c^2}$. Next, an affine transformation matrix $\mathbf{T}$ is formed as:

$$\mathbf{T} = \begin{bmatrix} \cos(\phi) + \alpha^2 \cdot (1 - \cos(\phi)) & \alpha \cdot \beta \cdot (1 - \cos(\phi)) & \beta \cdot \sin(\phi) & 0 \\ \alpha \cdot \beta \cdot (1 - \cos(\phi)) & \cos(\phi) + \beta^2 \cdot (1 - \cos(\phi)) & -\alpha \cdot \sin(\phi) & 0 \\ -\beta \cdot \sin(\phi) & \alpha \cdot \sin(\phi) & \cos(\phi) & d \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

where $\alpha = \frac{b}{\sqrt{a^2+b^2}}$, $\beta = \frac{-a}{\sqrt{a^2+b^2}}$, $\cos(\phi) = \frac{c}{\|\mathbf{z}\|}$ and $\sin(\phi) = \frac{\sqrt{a^2+b^2}}{\|\mathbf{z}\|}$. Then, the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ are easily extracted, given that $\mathbf{T} = [\mathbf{R}|\mathbf{t}]$. Finally, the transformed point $\mathbf{X_t}$ is obtained by taking:

$$\mathbf{X_t} = \mathbf{R}\mathbf{X} + \mathbf{t}, \quad (4)$$

and this transformation is applied to every point $\mathbf{X}$.

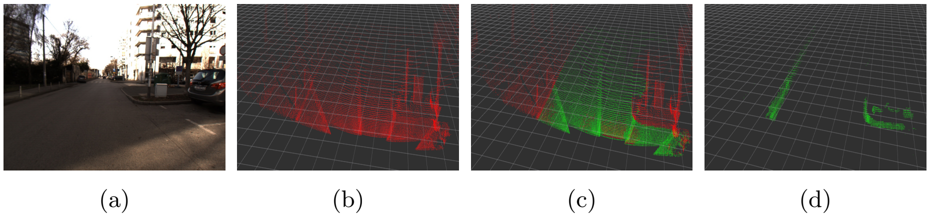
Next, geometric features from which the curb is determined are computed. Let $\mathbf{n}_p$ denote the normal vector at the considered 3D point. Given the ground plane normal $\mathbf{z}$, the angular distance $\theta$ is expressed as:

$$\theta = \arccos(\mathbf{n}_p \cdot \mathbf{z}), \quad (5)$$

with the direction of the normal $\mathbf{n}_p$ always oriented such that $\mathbf{n}_p \cdot \mathbf{z} > 0$, hence $\theta \in [0, \frac{\pi}{2}]$. Finally, the curb is determined using the computed angular distance $\theta$, by labeling a 3D point as a curb if $\theta$ falls in a predefined sub-interval, i.e., if $\theta \in [\frac{\pi}{8}, \frac{\pi}{4}]$ holds as proposed in [4]. The feature calculation process is done using the Point Could Library [8].

The process of normal estimation is heavily affected by the chosen search radius $R$. If set improperly, the feature calculation process degrades. When considering a 3D point, this radius defines an area from which the points are taken into account to fit a plane. Following manual inspection of the results on our data, we set $R$ to 0.3 m.

LiDAR-based curb detection is shown in Fig 4. The two leftmost images depict an exemplar scene as a color image (Fig 4a) and a LiDAR scan (Fig 4b). The estimated ground plane is shown in Fig 4c, while the curbs, extracted by using the angular distance feature are displayed in Fig 4d. Finally, by means of qualitative inspection, we observe that this method exhibits a higher false positive



(a) (b) (c) (d)

Fig. 4: Visualization of LiDAR-based curb extraction process via geometric features. First two images display sensor data; (a) RGB image (not used for computation, displayed for convenience), (b) LiDAR scan, which represent the scene. The following two images show; (c) ground-plane estimation, (d) estimated curbs.

rate (Fig 4d), in contrast to the described vision-based approach. However, we note that this method is fairly straightforward, so this comparison cannot serve as a conclusion regarding superiority of vision-based over LiDAR-based methods in general. Nonetheless, LiDAR is useful in poor lighting conditions. As future work, we plan to enhance this approach by introducing additional informative features, as well as filtering (e.g., a Kalman filter).

## 4 Camera-LiDAR calibration method

The major challenge of multi-sensor calibration lies in the fact that the only thing that can be surely measured in both sensor domains is the planar shape of a calibration checkerboard. The calibration process expects a set of synchronized camera images and LiDAR point clouds, where each pair contains measurements of the checkerboard. First, the parameters of the commonly used pinhole radtan camera model are determined. OpenCV functions for corner extraction and camera calibration are used to compute the intrinsic camera parameters and relative transformation from the calibration target reference frame to the camera reference frame $T_T^C$ for each frame in the measurement set. Furthermore, in each point cloud we seek to find the closest planar surface assuming it is the calibration target. We find this assumption reasonable since the calibration is performed in a controlled environment, and the ground plane can be easily filtered out by eliminating surfaces within a certain range around the ground plane normal vector.

The point cloud filtering step leaves us with points $p_L(i)$ corresponding to the checkerboard, expressed in the LiDAR reference frame. Given a valid LiDAR to camera transformation $T_L^C$, the point cloud is transformed from the LiDAR reference frame to the calibration target reference frame using the following transformation chain:

$$p_T(i) = (T_T^C)^{-1} \; T_L^C \; p_L(i). \tag{6}$$

Expressed in the target reference frame, all measured LiDAR points $p_T(i)$ should have a zero $Z$ coordinate. Extrinsic parameters estimation between the camera and the LiDAR is thus achieved by minimizing the following cost function:

$$T_L^C = \underset{T_L^C}{\arg\min} \left( \sum_i (T_T^C)^{-1} \; T_L^C \; p_L(i) \right)_Z . \tag{7}$$

## 5 Conclusion

In this paper, we have tackled the problem of estimating curbs in urban driving scenes for the purpose of controlling the brushes of a road sweeping utility vehicle. To this end, two different approaches were developed and qualitatively compared: a vision-based and LiDAR-based curb detection. Furthermore, we also described our data generation process – the camera-LiDAR rig preparation, data collection, and sensor setup calibration. As future work, we plan to investigate fusion of the vision-based and LiDAR-based approach, employ filtering on the LiDAR data, and explore learning-based approaches for curb detection in the LiDAR point cloud.

## References

1. Fernández, C., Izquierdo, R., Llorca, D.F., Sotelo, M.A.: Road curb and lanes detection for autonomous driving on urban scenarios. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). pp. 1964–1969 (2014)
2. Gao, R.: Rethink dilated convolution for real-time semantic segmentation. arXiv preprint arXiv:2111.09957 (2021)
3. Goga, S.E.C., Nedevschi, S.: 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). pp. 301–308. IEEE (2018)
4. Hervieu, A., Soheilian, B.: Road side detection and reconstruction using lidar sensor. In: 2013 IEEE Intelligent Vehicles Symposium (IV). pp. 1247–1252 (2013)
5. Hirschmuller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 2, pp. 807–814. IEEE (2005)
6. Hirschmüller, H.: Semi-global matching-motivation, developments and applications. Photogrammetric Week 11 pp. 173–184 (2011)
7. Kellner, M., Hofmann, U., Bouzouraa, M.E., Stephan, N.: Multi-cue, model-based detection and mapping of road curb features using stereo vision. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems. pp. 1221–1228 (2015)
8. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA). IEEE, Shanghai, China (May 9-13 2011)
9. Xu, J., Xiong, Z., Bhattacharyya, S.P.: Pidnet: A real-time semantic segmentation network inspired from pid controller. arXiv preprint arXiv:2206.02066 (2022)
10. Zhang, Y., Wang, J., Wang, X., Dolan, J.M.: Road-segmentation-based curb detection method for self-driving via a 3d-lidar sensor. IEEE transactions on intelligent transportation systems **19**(12), 3981–3991 (2018)