Redundant Multi-Object Detection for Autonomous Vehicles in Structured Environments

REDUNDANT MULTI-OBJECT DETECTION FOR AUTONOMOUS VEHICLES IN STRUCTURED ENVIRONMENTS

Stefano Feraco*, Angelo Bonfitto, Nicola Amati, Andrea Tonoli

Politecnico di Torino, Torino, Italy

*E-mail of corresponding author: stefano.feraco@polito.it

Resume

This paper presents a redundant multi-object detection method for autonomous driving, exploiting a combination of Light Detection and Ranging (LiDAR) and stereocamera sensors to detect different obstacles. These sensors are used for distinct perception pipelines considering a custom hardware/software architecture deployed on a self-driving electric racing vehicle. Consequently, the creation of a local map with respect to the vehicle position enables development of further local trajectory planning algorithms. The LiDAR-based algorithm exploits segmentation of point clouds for the ground filtering and obstacle detection. The stereocamera-based perception pipeline is based on a Single Shot Detector using a deep learning neural network. The presented algorithm is experimentally validated on the instrumented vehicle during different driving maneuvers.

Article info

Received 29 April 2021
Accepted 4 June 2021
Online 30 September 2021

Keywords:

perception,
autonomous driving,
obstacle detection,
point-cloud segmentation,
single shot detector,
LiDAR (Light Detection and Ranging)

Available online: https://doi.org/10.26552/com.C.2022.1.C1-C17

ISSN 1335-4205 (print version)
ISSN 2585-7878 (online version)

1 Introduction

Self-driving vehicles are experiencing a steadily increasing interest all over the world thanks to the most recent technological development, as witnessed in [1] and [2]. Although many Advanced Driving Assistance Systems (ADAS) are already present in the majority of vehicles of the recent mass production [3], fully autonomous vehicles are still considered as a disruptive force that could eruptively change the traffic environment and the whole mobility in the next years, thanks to the contribution of Artificial Intelligence, as stated in [4-6]. Recently, immense research efforts have been dedicated to autonomous systems and the DARPA Grand Challenge is one of the great results that the global mobile robotics community has achieved in the last decades [7-8], allowing to reach the high level of autonomy in nowadays commercial cars [9]. Moreover, about 94% of the road accidents are caused by human errors, according to a recent survey [10]. Therefore, these efforts have been motivated not only by the promise of preventing accidents, but also of reducing emissions and reducing driving-related stress [11]. Nevertheless, a consistent burden to adoption of driverless vehicles is the lack of public trust, since significant concerns, including but not limited to privacy and cybersecurity, have arisen [4]. Considering this framework, environment perception is a fundamental task for autonomous vehicles, which provides the vehicle a crucial assessment about the driving scenario, including an accurate information about the surrounding obstacles positions [12].

A vast variety of sensors is currently exploited for the purpose of environment perception with peculiar characteristics [13]. Cameras can offer a wide range of configurations in resolution, frame rate, size and optics parameters. Moreover, stereocameras are effective sensors in the self-driving vehicles since they can also be used to estimate depth map from images, enabling further obstacle detection and trajectory planning algorithms [14-15]. Nevertheless, camera-based sensors have some drawbacks for autonomous driving tasks under varying light and visibility conditions and with scenes with a high dynamic range, such as in entering or exiting a tunnel [13]. Furthermore, stereo vision is characterized by depth inaccuracies in the case of low-textured patterns [13] and the computed depth map has typically a limited range, which could be useless for automated driving at high speed. Therefore, the LiDAR (Light Detection and Ranging) sensors represent a recent technology that accurately computes distance to objects by measuring the flight-time of multiple laser light pulses [13]. Although the LiDAR sensors are mostly indicated devoted to the creation of accurate 3D maps in a huge horizontal Field-Of-View (FOV) [13, 16], they have also some important drawbacks in the environment perception. They typically have a limited vertical resolution and they are not suitable for detecting small objects placed at great distances since they compute a sparse map [13]. Moreover, the LiDAR measurement could be strongly affected by light and weather conditions, suggesting the usage of the redundant camera sensors in self-driving vehicles [17-18]. In addition, the Radar

sensors are used in the automotive perception tasks, being characterized by a strong measurement robustness with respect to the light and weather conditions. However, the high sensibility to target reflectivity and the low resolution of the radar technology strongly discourage the application of the technology for certain kinds of driving scenario, such as environments with small or far obstacles, which are detected with a low accuracy [13]. Therefore, the Radar sensors are not considered for the proposed sensing architecture.

This paper proposes a combined sensor architecture with both stereocamera and LiDAR sensors to enhance the perception pipeline robustness in terms of redundancy of the system. The redundancy in the proposed perception algorithm is of pivotal importance to avoid misclassification in the object detection process and poor environment sensing, as witnessed in the recent literature [19-21]. The investigated method does not implement a sensor fusion technique between the stereocamera and LiDAR, since it is intended to build a local map from the sensors, even in the case of a failure on one of the two sensors. This task is crucial to enable any further trajectory planning and control algorithm for autonomous driving [22-25].

Specifically, the proposed perception method is devoted to a driverless electric racing vehicle, thus requiring redundancy and robustness in the environment sensing process. The LiDAR-based perception pipeline relies on data coming from a Velodyne VLP-16 sensor that is placed onto the vehicle's front wing. The sensor can provide a full 360° view of the surrounding environment at 10 Hz to obtain an accurate real-time 3D data reconstruction, recorded by 16 light channels. It ranges up to 100 m with 30° vertical field-of-view (FOV) and an angular resolution up to 0.1° in the horizontal plane. The stereocamera-based perception algorithm exploits a Stereolabs ZED dual camera that is mounted on the top of the vehicle's roll bar. The stereocamera can perform a long-range 3D sensing up to 20 m distance with an increased accuracy in the short range (less than 10 m). The vehicle also features an NVIDIA Jetson AGX Xavier high-performance computing platform with embedded GPUs to process data coming from these sensors in a dedicated Robotic Operating System (ROS) software environment. The vehicle is autonomously driven on a racetrack at a varying speed without any prior knowledge of the path. The racetrack is properly structured with traffic cones, which are peculiar in terms of shape and colors (blue, yellow and orange). Since the ROS is not a hard-real-time system, a proprietary model-based software interface called RTMaps is used in the hardware that manages the vehicle dynamics control. In detail, RTMaps is a component-based software development and execution environment, which enables the synchronization and hard-real-time requirements for further control strategies that are implemented on target hardware machines.

The three-dimensional raw point cloud recorded by the LiDAR sensor is processed with a segmentation algorithm that is properly designed for the ground plane filtering. This filtering task is a common practice that is necessary to avoid considering ground points in the obstacles detection stage [26-27]. Therefore, an algorithm for point-cloud clustering is applied to detect clusters of point. Each cluster represents a detected object delimiting the structured track, as witnessed by other methods for clusters detection discussed in the recent literature [28]. The proposed LiDAR-based perception pipeline can estimate the distance and the position of the clusters representing the obstacles in the driving environment. In the autonomous driving framework, alternative methods for the object detection in 3D point clouds are voxel-based Artificial Neural Networks (ANNs) [29] and other architectures of deep convolutional ANNs [30-32]. The vision-based perception pipeline is based on an SSD (Single Shot Detector) architecture based on MobileNetV1 using a single deep learning neural network to perform the object detection task in each frame. The SSD-based perception algorithms are proven to be very fast and robust in the recent literature [33-35]. The proposed perception algorithm can accurately estimate the distance of the detected objects by means of matching the depth map generated from the ZED stereocamera with the bounding boxes identified in the image by the SSD. The information deriving from the LiDAR and stereocamera pipelines is fused and synchronized to compute a detailed local map with the sensed obstacles up 20 m in front of the vehicle. Nevertheless, each of the two pipelines is redundant with respect to the other one in order to prevent inaccuracies in the obstacle detection process. Creation of a resulting local map enables further trajectory planning algorithms.

Therefore, the main contribution of this manuscript is to provide a redundant combined method for the multi-object perception in a structured environment for an autonomous electric racing vehicle. Moreover, the peculiar design and integration of the perception pipeline is tested during the extensive experimental validation on a real vehicle. The reported results include a set of different outdoor driving situations at a varying vehicle's speed. The redundancy in the perception pipeline is not novel at a system level in the context of driverless racing competitions, since it has been already proposed in [19] and [36]. Nevertheless, the investigated scheme is novel with respect to the existing literature, since it is based on an SSD and a clustering algorithm for point clouds that work in parallel, thus allowing to have a fully independent throughput from the perception pipeline. This configuration has not been reported in the literature so far.

The paper is structured as follows: section 2 illustrates the design of the proposed obstacle detection method for the LiDAR-based and stereocamera-based algorithms, along with the considered vehicle setup and the retained hardware and software architecture; section 3 presents the obtained results for both the perception pipelines and the creation of the local map during different maneuvers.

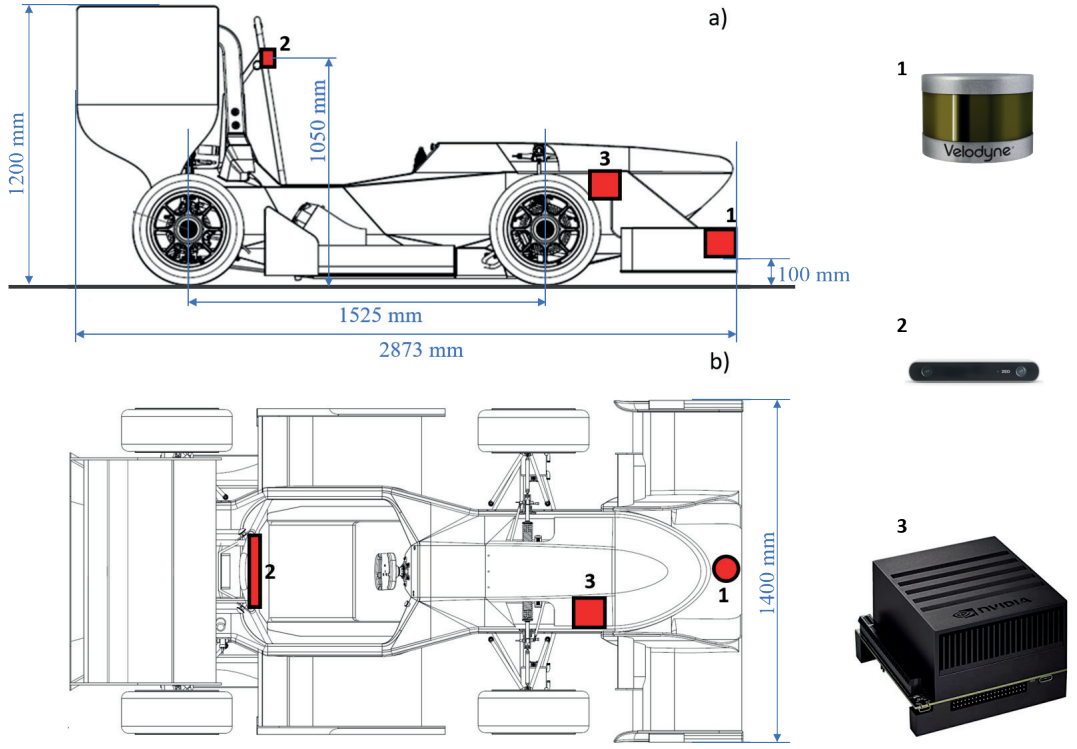

Figure 1 *Vehicle and considered hardware positions: a) side view; b) top view. 1: Velodyne VLP-16 LiDAR sensor; 2: Stereolabs ZED stereocamera; 3: NVIDIA Jetson AGX Xavier high-performance computing platform*

2 Method

In this section, the retained vehicle setup is presented first. Furthermore, the complete hardware and software architecture, deployed on a vehicle, is illustrated. Eventually, the designed LiDAR-based and stereocamera-based perception methods are described in devoted subsections, respectively, providing pseudo-code of the implemented algorithms.

2.1 Vehicle layout and hardware/software setup

The considered all-wheel drive electric vehicle is represented in Figure 1. A high-performance racing vehicle is considered due to reasons of prototyping, however, the outcomes of the research paper can be easily adapted to a commercial vehicle. The vehicle has an integral carbon fiber chassis built with honeycomb panels, double wishbone push-rod suspensions, an on-wheel planetary transmission system and a custom aerodynamic package. The vehicle can reach a maximum speed equal to 120 km/h with longitudinal acceleration peaks reaching up to 1.6 g. The Velodyne VLP-16 LiDAR sensor is mounted on the front wing of the vehicle. The sensor is fixed at a height equal to 0.1 m from the ground. The Stereolabs ZED stereocamera sensor is mounted at a height of 1.05 m from the ground and it is fixed to the vehicle's rollbar, as represented in Figure 1. The NVIDIA Jetson Xavier high-performance computing platform is placed inside the vehicle's monocoque, fixed to its right side. The main vehicle's parameters are listed in Table 1. A proper wiring system has been set up to correctly interface and supply the sensors to the computing platform, using a 12 V 10 Ah rechargeable Lithium battery as the devoted power source along with properly connected DC-DC power converter for the computing platform.

The Velodyne VLP-16 LiDAR sensor provides a full 360-degree point-cloud of the surrounding environment at 10 Hz to obtain an accurate real-time data reconstruction recorded by 16 light channels. It ranges up to 100 m with 30° vertical field-of-view (FOV) and an angular resolution up to 0.1° in the horizontal plane [37]. The LiDAR sensor is connected to the computing platform with embedded GPUs through an Ethernet connection. Specifically, the computing platform creates a ROS network, which allows to process the information streaming from the LIDAR-based sensor. The ROS Melodic (2018) is used for the arm64 architecture of the computing platform that features Ubuntu 18.04 release.

The Stereolabs ZED stereocamera is connected via 3.0 USB port to the computing platform. The considered stereocamera features stereo 2K cameras with dual 4MP RGB sensors. It has a field of view of 110° and can stream uncompressed video at a rate up to 100 FPS. Left and right video frames are synchronized and streamed as a single uncompressed video frame format. Several configurations parameters, as resolution, brightness, contrast, saturation, can be tuned properly [38]. In the investigated algorithm, default parameters have been set both for the LiDAR and stereocamera sensors. The camera is used in the high-

Table 1 *Main parameters of the considered all-wheel drive electric vehicle*

Parameter	Symbol	Value	Unit
Mass	m	190	kg
Vehicle wheelbase	l	1.525	m
Overall length	L	2.873	m
Front axle distance to CG	a	0.839	m
Rear axle distance to CG	b	0.686	m
Vehicle track width	t	1.4	m
Overall width	W	1.38	m
Wheel radius	R_W	0.241	m
Maximum power	P_{max}	80	kW
Motors peak torque	T_{max}	84	Nm

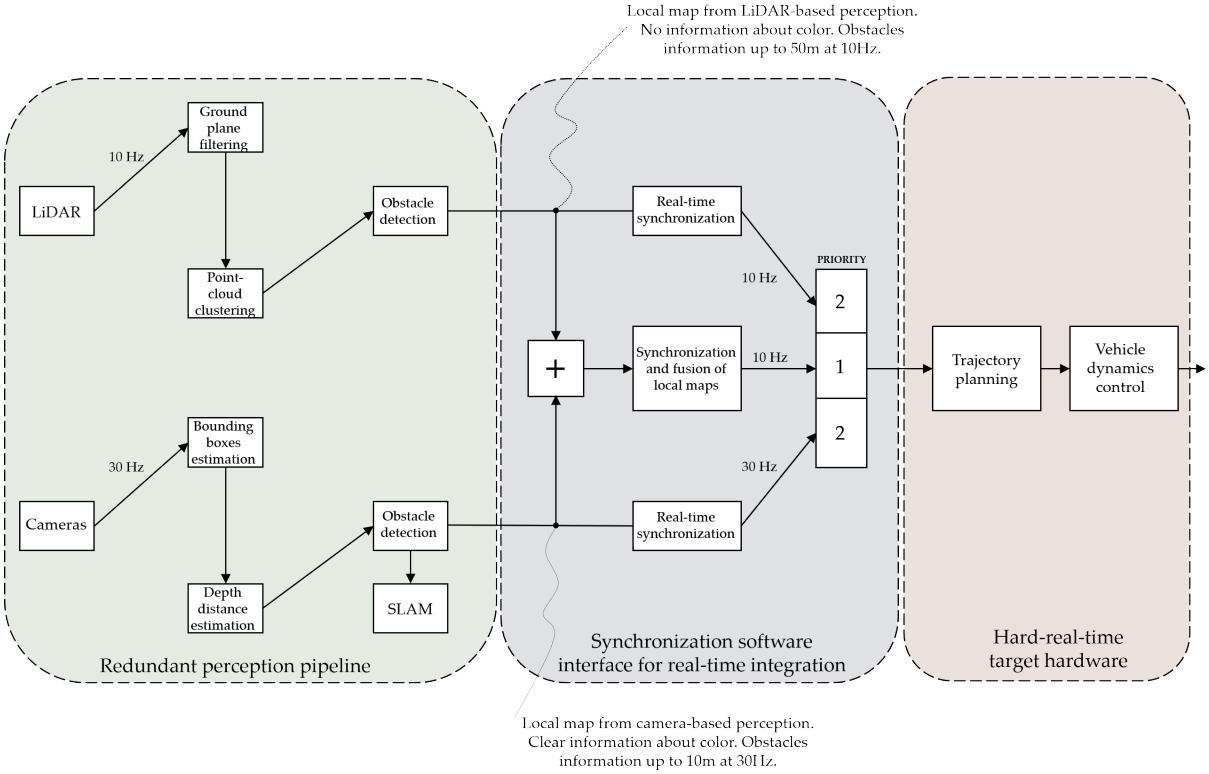

Figure 2 *Redundant perception pipeline and global software architecture*

definition 1080 mode (HD1080) at 30 Frame Per Second (FPS). ZED stereocamera is used since it is capable of accurately recording dense depth map information using triangulation from the geometric model of non-distorted rectified cameras up to 10 m [38].

The NVIDIA Jetson AGX Xavier is an embedded Linux high-performance computing platform with embedded GPUs with 32 TOPS of peak computational power and 750 Gbps of high-speed input/output capability in less than 50 W of needed power. The retained high-performance computing platform enables intelligent vehicles with end-to-end autonomous capabilities since it is based on the most complex System-on-Chip (SoC) ever created up to 2018 [39]. The platform comprises an integrated 512-core NVIDIA Volta GPU including 64 Tensor Cores, 8-core NVIDIA Carmel ARMv8.2 64-bit CPU, 16GB 256-bit LPDDR4x, dual NVIDIA Deep Learning Accelerator (DLA) engines, NVIDIA Vision Accelerator engine, HD video codecs, 128 Gbps of dedicated camera ingest and 16 lanes of PCIe Gen 4 expansion. Memory bandwidth over the 256-bit interface weighs in at 137 GB/s, while the DLA inference accelerator engines offload inferencing of Deep Neural Networks (DNNs). The NVIDIA's JetPack Software Development Kit 4.1.1 deployed for Jetson AGX Xavier includes CUDA 10.0, cuDNN 7.3 and TensorRT 5.0 libraries, thus providing a complete artificial intelligence software stack [39].

In order to validate the proposed method, the driving environment is properly structured with traffic cones according to the rules listed in [40]. In fact, each traffic cone has a height equal to 0.325 m and a square base, with a side length equal to 0.228 m. The cones of the right lane boundary are yellow with a black stripe, while the right

lane boundary is built with blue cones with a white stripe. Bigger orange cones indicate the starting and the ending points of the track.

The sensor fusion is a renowned solution to increase obstacle estimation accuracy [36, 41], thus it could be also used as an alternative approach for the investigated application. Nevertheless, the sensor fusion is always constrained by the sensor with lower throughput frequency, i.e. 10 Hz in this case and is strongly dependent on both used sensors. On the contrary, as represented in Figure 2, the proposed redundant perception pipeline can provide consistent results even in the case of failure of a single sensor. Moreover, the information at 30 Hz from camera-based pipeline is used for the mapping purposes, at the same time. Furthermore, a fused local map with higher priority for trajectory planning is created in the real-time synchronizing interfaced (RTMaps) at 10 Hz rate. This approach is quite different from the sensor fusion at a sensor level, that is performed in [41], because the information is fused and synchronized in the local map building process via over-imposition and synchronization of the detected obstacles. Indeed, avoiding the sensor fusion at a sensor level can save computational costs, as no additional algorithms are deployed on the devoted control unit. In fact, fusing information on the created local map only involves an easy operation of two-dimensional points over-imposition and time synchronization. The obtained local map is thus used with the higher priority for the local trajectory planning algorithms.

2.2 LiDAR-based perception algorithm

The LiDAR sensor records point clouds at a frequency equal to 10 Hz consisting of thousands of 3D points in a 360° range on the horizontal plane, while the vertical FOV is ±15°. Each point-cloud contains the distance of each point in the 3D space along with the intensity of the reflected light in that point. Then, the raw point cloud is filtered by removing all the points out of the region-of-interest (ROI). Therefore *ROI* (*x, y, z*) is defined as:

$$ROI(x,y,z) = \left\{ \begin{array}{c} 0 < x < 50 \\ -15 < y < 15 \\ -0.5 < z < 1.5 \end{array} \right\}, [m]. \quad (1)$$

A ground plane filtering segmentation algorithm is then applied to the raw point-cloud in *ROI* (*x, y, z*) in order to remove all the points belonging to the ground, which can badly affect the proposed object detection method. Therefore, a clustering algorithm is applied the filtered point-cloud and the distance to the detected obstacles is finally estimated.

2.2.1 Ground-plane filtering segmentation

Efficient segmentation algorithms often try to reduce dimensionality from LiDAR 3D point-clouds to 2D grid fixed to the ground. Firstly, this approach was proposed by many teams participating at DARPA Urban Challenge robot competition in past years [42-44], although it is heavily affected by under-segmentation issues and merging of different objects in the same segments [26]. Recently, other algorithms can process the point-cloud in full 3D [26, 45-49]. However, most of them could not be implementable in real-time because of the large amount of points they take into account resulting into a high computational costs, which can be solved only by means of treating the point cloud in cylindrical coordinates and fitting the line segments to the point-cloud [26]. Considering a LiDAR 3D point-cloud, the points belonging to the ground surface are often the majority of the recorded points. Therefore, the removal of the ground points can reduce the computational efforts required by further algorithms and ease the object detection task. In the proposed algorithm, an iterative multiple plane fitting technique for the ground-plane filtering is applied to the LiDAR 3D point-cloud in the considered *ROI* (*x, y, z*). A single plane model is often inaccurate in the segmentation of the actual ground surface, since the ground points are not uniquely defined and the LiDAR measurements are often affected by huge noise, especially in the case of long distances [27].

The investigated algorithm divides the point-cloud into multiple segments along the direction of travel of a vehicle, namely the x-axis and iteratively applies the ground plane filtering algorithm in each of these segments. In each point-cloud, the retained LiDAR sensor can measure 16 segments according to the 16 sensing channels. The proposed ground plane filtering algorithm extracts a set of points with low height values (seeds), which are used to estimate the plane model of the ground surface in each of the considered point-cloud segments. The initial seed points are defined using the lowest point representative (LPR) approach, using an average value of the lowest height values of points in the point-cloud [27]. Therefore, noisy measurements do not affect the plane estimation. Once the LPR is computed, it is assumed as the point with the lowest height in the point cloud, thus all the points inside the seed height threshold are used to build the initial seeds set. The seed height threshold is heuristically set to 0.35 m in the proposed algorithm. Then, each point in the segment is evaluated with respect to the estimated plane model by computing the distance from the orthogonal projection of the point onto the identified plane to the point itself. The resulting distance is then compared to a threshold distance, which is heuristically defined to decide if the point belongs to the ground surface or not. The points belonging to the ground plane are used as new seed points to estimate a new plane model and the process repeats for the defined number of iterations. Eventually, all the identified ground points in each of the segments are concatenated to define the ground plane. The number of iterations is heuristically set to 80, the threshold distance is set to 0.1 m and the number of LPR initial seeds is set to 14. At each iteration, the ground plane is estimated by a simple linear model that is defined

as follows:

$$ax + by + cz + d = 0\,, \tag{2}$$

$$\tau^T x = -d\,, \tag{3}$$

where $\tau^T = [a\,b\,c]^T$ and $x = [x\,y\,z]^T$ and solved for τ by using the covariance matrix $C \in R^{3\times3}$ that is computed considering the set of seed points $S \in R^3$ as in:

$$C = \sum_{i=1:|S|}(s_i - \hat{s})(s_i - \hat{s})^T\,, \tag{4}$$

with $\hat{s} \in R^3$ that is the mean of all the seeds $s_i \in R^3$. The covariance matrix C should be decomposed in three singular vectors describing the directions of the seed dispersion by applying the singular value decomposition. Then, considering the ground plane, the normal τ indicates the direction with the least variance. Therefore, d can be computed using Equation (2) by substitution of x by $\hat{s}$. A complete theoretical background of the proposed method can be found in [27]. Once the ground plane has been identified, it can be filtered out of the point-cloud in order to consider only the remaining points for further processing algorithms. The pseudocode of the proposed approach is illustrated in Algorithm 1, that is represented in Figure 3.

Algorithm 1: Ground plane filtering for one segment of the point-cloud.

```
        INITIALIZATION
1  π_gnd: ground points
2  π_NOTgnd: non-ground points
3  Π: input point-cloud
4  Π_sortedH: input point-cloud sorted on height
5  μ_IT: number of iterations
6  μ_LPR: number of points used in the LPR estimation
7  Δ_seeds: initial seeds threshold
8  Δ_dist: plane distance threshold
        MAIN
9  π_gnd = InitialSeedsExtraction (Π, μ_LPR, Δ_seeds)
10 for i=1:μ_IT do
11      plane = EstPlane(π_gnd)
12      clear(π_gnd, π_NOTgnd)
13      for j = 1: |Π| do
14          if plane (p_k) < Δ_dist then
15              π_gnd ← p_k
16          else
17              π_NOTgnd ← p_k
18          end
19      end
20 end
21 InitialSeedsExtraction:
22 LPR = Average(Π_sortedH(1: μ_LPR))
22 for i = 1: Π do
23      if p_k.height < LPR.height + Δ_seeds then
24          seeds_set ← p_k
25      end
26 end
27 return (seeds_set)
```

Figure 3 *Algorithm 1 - ground plane filtering for one segment of the point-cloud*

2.2.2 Point-cloud clustering and obstacles distance estimation

At each iteration, the point-cloud, obtained after the ground-plane filtering process, is further segmented in order to detect significant clusters of points representing the objects in the structured environment. Positions of the clusters are then estimated in real-time.

The input of the cluster detector is the filtered point-cloud Π_{filt} that includes all the non-ground points π_{NOTgnd} defined in Algorithm 1. Non-overlapping clusters of adjacent points are then extracted considering their relative Euclidean distance in the three-dimensional space as commonly intended in [50] and [51]. However, considering the vertical angular resolution γ of the Velodyne VLP-16 LiDAR sensor equal to 2°, the resulting distance in the z-axis (vertical) can be huge for far objects. Although fast and effective, distance-based clustering could be inaccurate, especially in the case of distant objects. If the threshold distance is not properly set in the clusters definition, there is a risk of splitting single objects into multiple adjacent clusters or merging different objects into a single cluster [16]. Therefore, considering the two generic clusters Γ_α and Γ_β both included in Π_{filt}, the non-overlapping condition can be written as:

$$\Gamma_\alpha \cap \Gamma_\beta = \emptyset \Rightarrow \min \left\| \pi_{NOTgnd\alpha} - \pi_{NOT_{gnd\beta}} \right\| \geq \delta , \tag{5}$$

where $\alpha \neq \beta$ (i.e. the clusters are different), $\pi_{NOTgnd\alpha} \in \Gamma_\alpha$, $\pi_{NOTgnd\beta} \in \Gamma_\beta$ and δ is the threshold distance to define the cluster that is defined as follows:

$$\delta = 2 \cdot \rho \cdot \tan \frac{\gamma}{2} . \tag{6}$$

By considering the non-ground points π_{NOTgnd} only in the defined *ROI*(*x, y, z*), the risk of considering too many 3D points in the clustering is limited, avoiding to increase the computational effort. To further reduce the data complexity, the horizontal space in the xy-plane is divided into multiple nested regions at a fixed constant distance threshold within each of them, as proposed in [52]. Therefore, a set of threshold distance values δ_i is retained at multiple fixed intervals d_δ where $\delta_{i+1} = \delta_i + d_\delta$. In each of the defined intervals, the maximum cluster detection range ρ_i is computed using Equation (6). Therefore, the corresponding radius $\mathcal{R}_i$ is determined straightforward. The width of the i i-th region w_i is simply computed as:

$$w_i = \mathcal{R}_i - \mathcal{R}_{i-1} . \tag{7}$$

To define the circular region, a width w equal to 1.5 m is set heuristically in the proposed algorithm and d_δ is set to 0.1 m, after a necessary trial and error stage to find the best parameters for detecting traffic cones used in the retained structured environments. Therefore, clusters that are too large or too small are neglected.

Once the number of clusters k in Π_{filt} is computed, a straightforward implementation of the proposed clustering method can be obtained with the renowned k-d tree algorithm in order to group the points in the correct cluster [52-53]. An efficient alternative method can be k-means clustering [16, 54]. An analysis on the robustness and stability of a k-d tree implementation of the proposed method, dedicated to detection of humans in point-clouds, is given by [52].

For each of the detected clusters, representing the obstacles in the structured environment, the centroid of the cluster is considered as the position of a cone in the defined *ROI*(*x, y, z*), by means of collapsing all the clustered points onto the horizontal plane. This operation is performed by means of setting a null z-coordinate to the clustered points. Thus, the centroid of each cluster is computed as the geometric center of all the points in the cluster. Therefore, if a generic cluster Γ includes a certain number π_Γ of sparse points, its centroid $\omega(x_\omega, y_\omega)$ will be computed as:

$$\omega(x_\omega, y_\omega) = \left(\frac{x_1 + x_2 + \cdots + x_{\pi\Gamma}}{\pi_\Gamma}; \frac{y_1 + y_2 + \cdots + y_{\pi\Gamma}}{\pi_\Gamma} \right). \tag{8}$$

Eventually, a two-dimensional local map can be created considering the identified centroids. which represents the detected obstacles with respect to the LiDAR sensor position in the xy-plane.

2.3 Stereocamera-based perception algorithm

The stereocamera-based perception algorithm is designed to detect cones and extract the color features from the detected obstacles, namely blue, yellow and orange cones. The distance with respect to the sensor is then computed by matching the detected bounding boxes representing the obstacles with the recorded depth map from the ZED stereocamera. This algorithm is redundant to the LiDAR-based one. Nevertheless, it performs a peculiar task since it can estimate not only the position of the detected obstacles but also the color of the detected cones if nearer than 10 m from the sensor, i.e. the maximum distance in which the generated depth map is reliable. In this section, the proposed SSD design and architecture are presented. Then, the image and depth map matching method for distance estimation is discussed.

2.3.1 Single Shot Detector design and architecture

The proposed perception stereocamera-based algorithm exploits an SSD algorithm based on the renowned MobileNetV1 structure to detect the objects [55]. This Convolutional Neural Network (CNN) structure is used since it is proven to be accurate and very fast in the object detection task in the recent literature [56], thus being compliant with the proposed real-time application of autonomous driving. As a matter of fact, the proposed

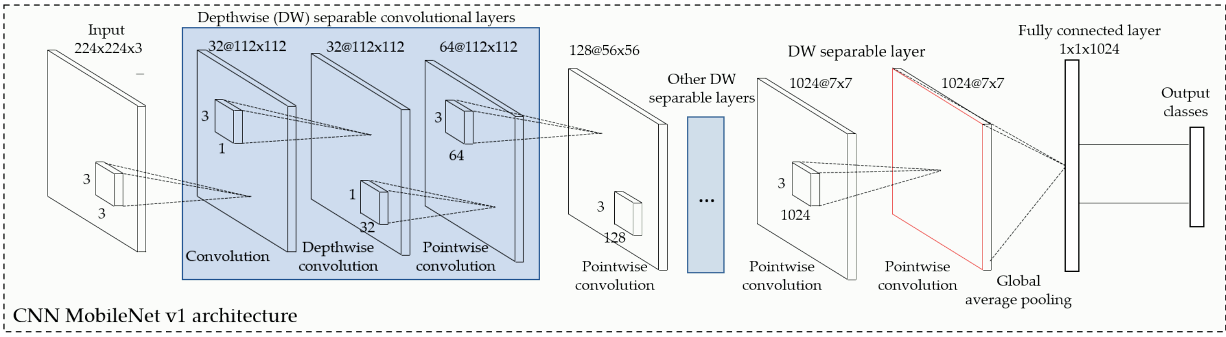

Figure 4 *CNN MobileNet v1 architecture*

Table 2 *MobileNetV1 CNN architecture. Conv: standard convolutional layer; Conv DW: depthwise convolutional layer; AVG Pool: average pooling layer*

Type	Stride	Kernel shape	Input size
Conv	2	3 × 3 × 3 × 32	224 × 224 × 3
Conv DW	1	3 × 3 × 32	112 × 112 × 32
Conv	1	1 × 1 × 32 × 64	112 × 112 × 32
Conv DW	2	3 × 3 × 64	112 × 112 × 64
Conv	1	1 × 1 × 64 × 128	56 × 56 × 64
Conv DW	1	3 × 3 × 128	56 × 56 × 128
Conv	1	1 × 1 × 128 × 128	56 × 56 × 128
Conv DW	2	3 × 3 × 128	56 × 56 × 128
Conv	1	1 × 1 × 128 × 256	56 × 56 × 128
Conv DW	1	3 × 3 × 256	28 × 28 × 256
Conv	1	1 × 1 × 256 × 256	28 × 28 × 256
Conv DW	2	3 × 3 × 256	28 × 28 × 256
Conv	1	1 × 1 × 256 × 512	14 × 14 × 256
Conv DW	1	3 × 3 × 512	14 × 14 × 512
Conv	1	1 × 1 × 512 × 512	14 × 14 × 512
Conv DW	1	3 × 3 × 512	14 × 14 × 512
Conv	1	1 × 1 × 512 × 512	14 × 14 × 512
Conv DW	1	3 × 3 × 512	14 × 14 × 512
Conv	1	1 × 1 × 512 × 512	14 × 14 × 512
Conv DW	1	3 × 3 × 512	14 × 14 × 512
Conv	1	1 × 1 × 512 × 512	14 × 14 × 512
Conv DW	1	3 × 3 × 512	14 × 14 × 512
Conv	1	1 × 1 × 512 × 512	14 × 14 × 512
Conv DW	2	3 × 3 × 512	14 × 14 × 512
Conv	1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv DW	2	3 × 3 × 1024	7 × 7 × 1024
Conv	1	1 × 1 × 1024 × 1024	7 × 7 × 1024
AVG Pool	1	Pool 7 × 7	7 × 7 × 1024
Fully Connected	1	1024 × 1000	1 × 1 × 1024
Softmax	1	Classifier	1 × 1 × 1000

vision-based algorithm can detect obstacles in the fed images and draw the related bounding boxes at a frequency up to 30 Hz in the actual real-time application when deployed on the NVIDIA Jetson AGX Xavier. Specifically, the left camera is used for the SSD object detector only. Once a sufficiently large dataset of images, related to the structured environment, has been collected, containing labeled images of the three classes of cones in different light conditions and weather, a proper transfer learning stage is applied in order to define the three retained object classes:

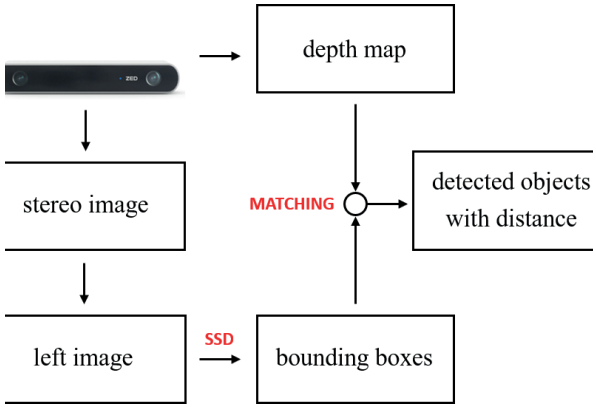

Figure 5 *Block-scheme of the proposed stereocamera-based perception algorithm*

blue cone (class 1), yellow cone (class 2) and orange cone (class 3). This is a standard procedure to apply a known CNN structure to a specific target domain of images, as widely discussed in [57].

MobileNetV1 is an efficient CNN architecture that use the depthwise separable convolutions, which factorize a standard convolution into a depthwise convolution and a pointwise convolution, in order to efficiently build lighter models with respect to earlier architectures [55]. Moreover, MobileNetV1 introduces two global hyper-parameters, which allow to perform a trade-off between latency and accuracy, namely the width multiplier and resolution multiplier. Therefore, the MobileNetV1 is built with multiple depthwise separable convolution layers and each depthwise separable convolution layer consists of a depthwise convolution and a pointwise convolution. The MobileNetV1 has 28 layers by counting depthwise and pointwise convolution as separate layers [55]. The size of the input images is $224 \times 224 \times 3$ pixels, thus the streaming images from the ZED left camera are properly resized before feeding them into the proposed CNN. The detailed architecture of the MobileNetV1 is given in Table 2 [55]. A complete theoretical background about the CNN MobileNetV1 can be found in [55].

Figure 4 illustrates the layout of the retained CNN MobileNetV1 with reference to Table 2 and [55].

2.3.2 Stereo image and depth map matching for distance estimation

After a preliminary camera calibration stage performed with the provided ZED Software Development Kit (SDK), the ZED stereocamera is ready to provide a reliable depth information in real-time up to 10 m distance, thanks to its embedded algorithms. The ZED stereocamera already provides rectified images, facilitating the stereo disparity estimation, which is a fundamental process prior to the estimation of the depth map [38]. The ZED camera can compute depth map using triangulation from the geometric model of non-distorted rectified cameras. The depth D of each point p is computed as:

$$D = \frac{fb}{xi^L - xi^R}, \tag{9}$$

where f is the focal length, b is the baseline distance of the stereocamera and $xi^L - xi^R$ is the disparity value [38, 58]. The focal length f is assumed equal for the two cameras retaining that they are co-planar with parallel optical axes. The left camera is assumed as the origin frame for the resulting depth map. Disparity map is inversely proportional to the two-dimensional depth map, since the high disparity means that the point is closer to the stereocamera baseline and vice versa.

The 3D reconstruction phase uses the depth information in the disparity map along with camera calibration parameters by matching the RGB pixels with the two-dimensional coordinates, related to the disparity map created with respect to the optical center of the left camera. The result of this process is a dense map of the RGB points in 3D coordinates [59], which is accurately obtained for distances lower than 10 m. Therefore, the obtained 3D reconstruction can be finally exploited for estimating the distance of the objects, corresponding to the identified bounding boxes. This task is commonly performed by computing the center point in each of the bounding boxes and projecting it into the disparity map. The distance from the left camera frame to each of the point is computed straightforward, as in [59] and [60]. Therefore, a two-dimensional local map can be computed by knowing the position and distance of the detected obstacles.

A block scheme of the proposed stereocamera-based perception algorithm is provided in Figure 5.

3 Results and discussion

In this section, the results of the proposed LiDAR-based and stereocamera-based object detection algorithms are presented, considering multiple dataset recorded on the instrumented vehicle in the structured environment. Specifically, the acquisitions are performed in Italy (Piedmont region) in 2020, in two different racetracks under different light and weather conditions. The proposed redundant method accurately detects the obstacles at fast refresh rates: up to 10 Hz for the LiDAR-based algorithm and up to 30 Hz for the stereocamera-based method, when running at the same time in the NVIDIA Jetson

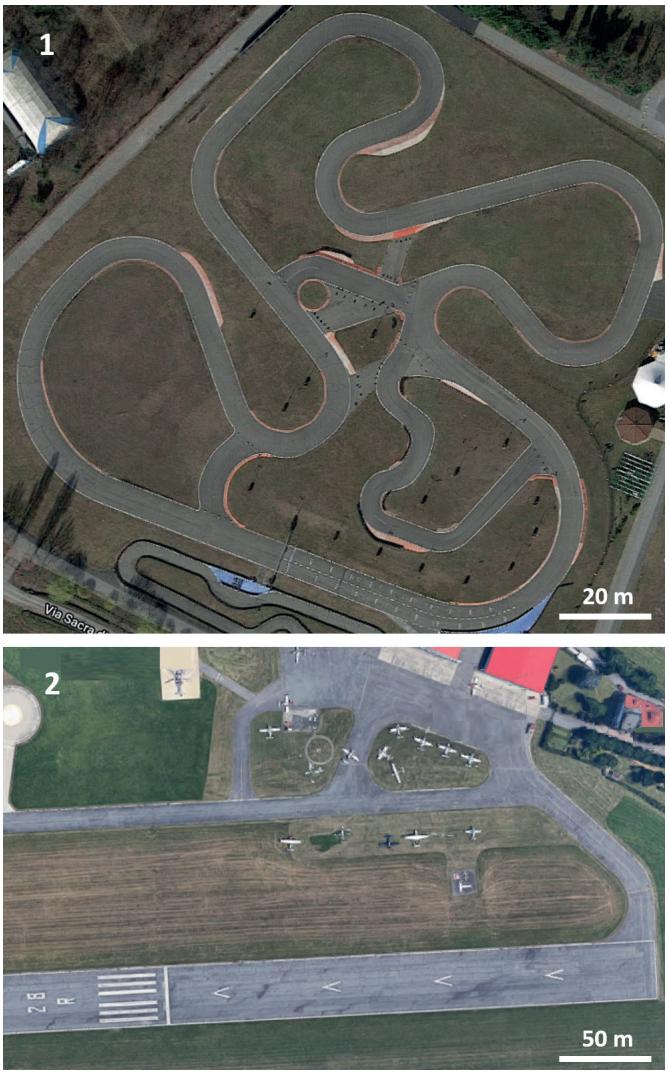

Figure 6 *Aerial views of a racetrack 1 and a racetrack 2 (data from Google Maps, 2020)*

AGX Xavier platform. The 10 Hz frequency of the LiDAR detections could not be feasible for the real-time assessment of environment perception during the vehicle motion at high speed. However, it is important to notice that the proposed perception method is particularly devoted to an accurate assessment of the obstacles position during the first lap of the racing track, that is driven at low speed (up to 15 km/h). Once the first lap is concluded, a global map can be generated via Simultaneous Localization and Mapping (SLAM) algorithms. Then, global trajectory planning algorithms can be implemented, neglecting the streaming data from the perception pipeline.

Figure 6 shows aerial images of the retained racetracks where the cones are properly placed to define the structured environment. The image data are taken from Google Maps service (2020).

Figure 7 illustrates an example of a single frame while moving in the racetrack 1. The LiDAR raw point-cloud is represented in Subfigure 7a and the ground-filtered point-cloud is shown in Subfigure 7b. LiDAR points have the color of the reflected light intensity parameters (from 0 to 255). In the figure, each square box has a side length equal to 1 m. The zoomed portions show how the unnecessary ground points are filtered out of the point-cloud without loss of information with respect to the points representing the obstacles.

Figure 8 represents the LiDAR-based detection results obtained at standstill in racetrack 1, at the starting point of the track. Clusters centroids are indicated with red arrows in Subfigure 8b and the LiDAR sensor position is indicated by a black dot. The resulting two-dimensional local map is shown in Subfigure 8b with by grey dots representing the centroids of the detected obstacles.

Similarly, Figure 9 illustrates the LiDAR-based detection results obtained at standstill in racetrack 1, at the end of the track. Clusters centroids are indicated with red arrows in Subfigure 9b and the two-dimensional local map is shown in Subfigure 9b.

Figure 10 represents the detection results obtained with the LiDAR-based algorithm in poor weather and light conditions in racetrack 2, with the vehicle moving in a left curve. The vehicle speed is not constant during this maneuvers and can reach up to 80 km/h. Clusters centroids are indicated with red arrows in Subfigure 10b and the two-

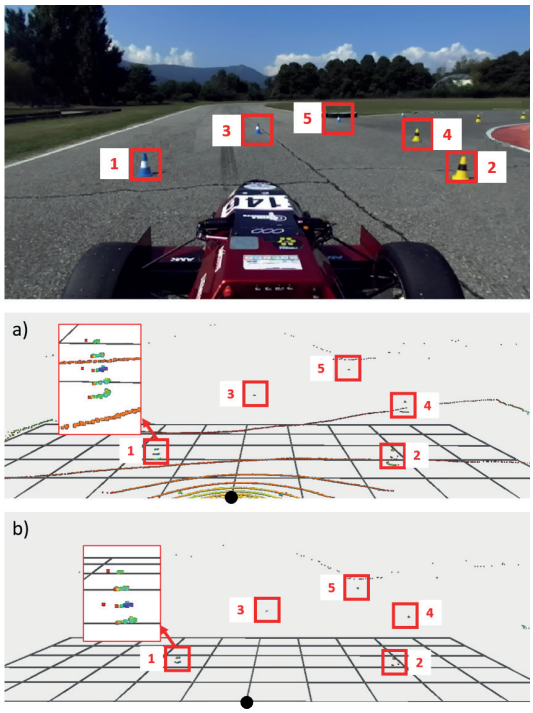

Figure 7 *Example of a single LiDAR frame while moving in the driving scenario (racetrack 1): a) raw point-cloud; b) filtered point-cloud*

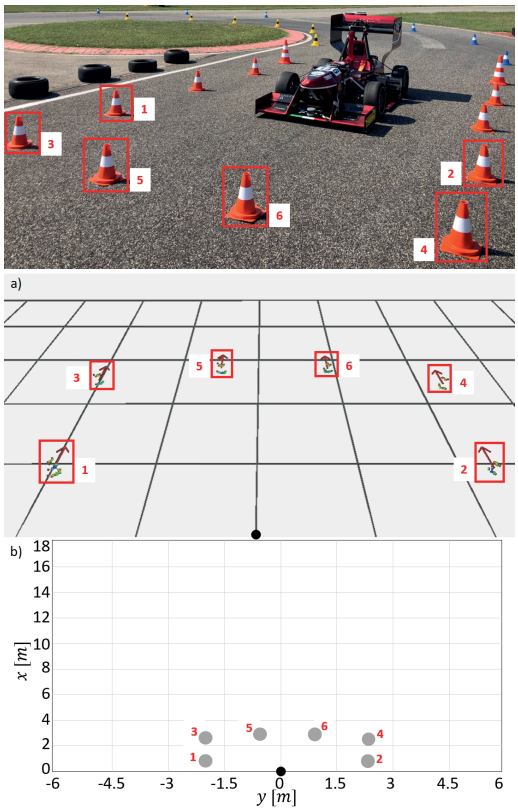

Figure 9 *LiDAR-based detection results (racetrack 1) at standstill (end of the track): a) filtered ground-points and cluster centroids; b) resulting 2D local map*

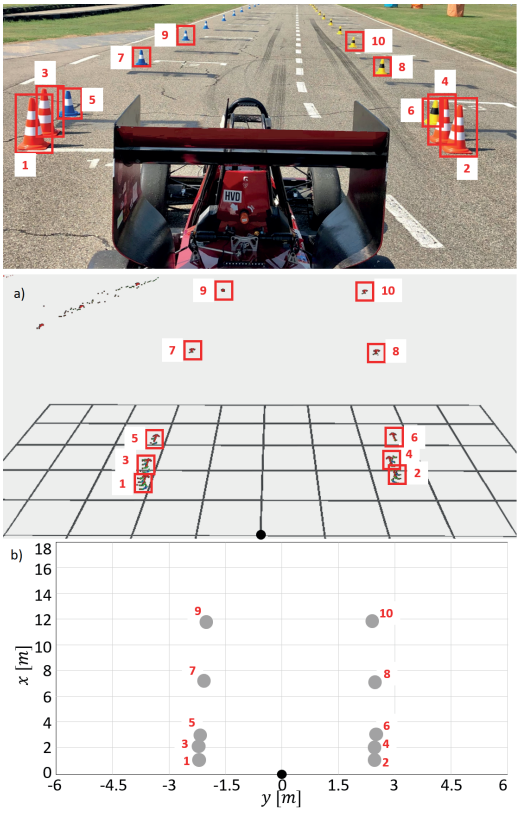

Figure 8 *LiDAR-based detection results (racetrack 1) at standstill (start of the track): a) filtered ground-points and cluster centroids; b) resulting 2D local map*

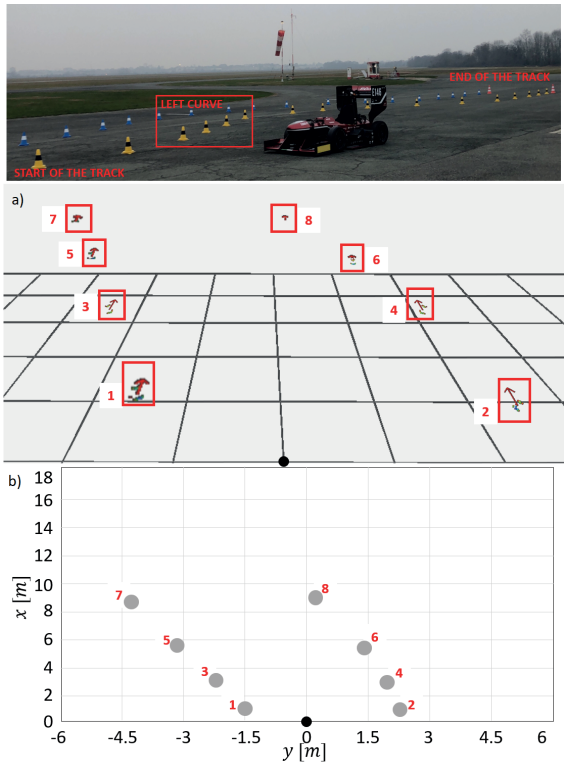

Figure 10 *LiDAR-based detection results (racetrack 2) during the cornering (left curve): a) filtered ground-points and cluster centroids; b) resulting 2D local map*

dimensional local map is shown in Subfigure 10b. Similarly, Figure 11 illustrates the results obtained in the same poor light and weather conditions (racetrack 2), while the vehicle is moving in a right curve.

Figures 12-14 illustrate the results obtained with the stereocamera-based algorithm along with the resulting two-dimensional local map in the following maneuvers: at standstill at the start of the racetrack (Figure 10); approaching the end of the racetrack 2 (Figure 13); during the cornering left or right in racetrack 1 (Figures 14 and 15, respectively). The sensor position in the map is represented by a black dot, while the other dots represent the estimated position and color for each of the detected obstacles. The computed bounding boxes and the estimated obstacles distances are reported in each of Figures 12-15 onto the recorded raw left image of the stereocamera.

The proposed redundant perception method is not affected by ambient and light conditions since it is proven in different racetracks and weather circumstances. The ground points are correctly filtered out of the raw point-cloud and the information related to the obstacles points is robustly preserved from being filtered. The refresh map of the two-dimensional local map is always high when deployed on the NVIDIA Jetson AGX Xavier platform, thus

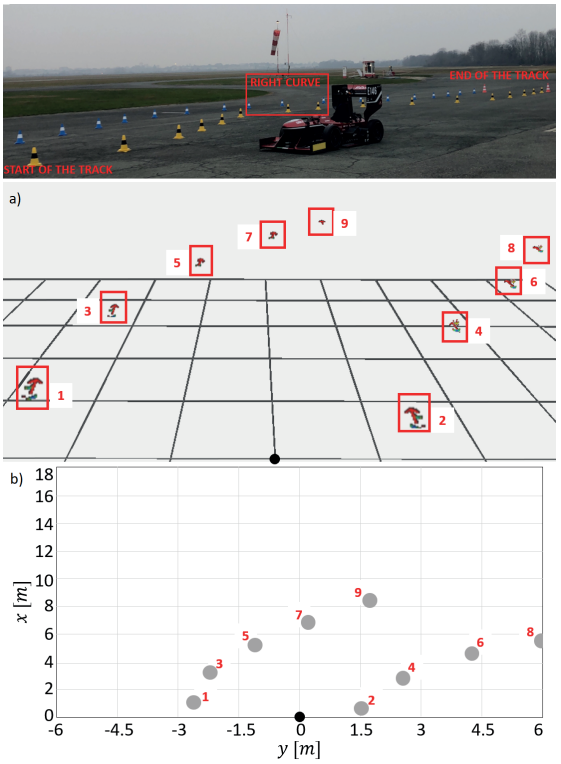

Figure 11 *LiDAR-based detection results (racetrack 2) during the cornering (right curve): a) filtered ground-points and cluster centroids; b) resulting 2D local map*

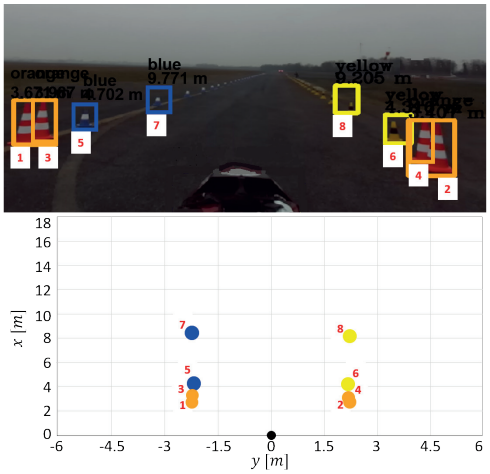

Figure 12 *Stereocamera-based detection results and resulting 2D local map (racetrack 2) at standstill (start of the track)*

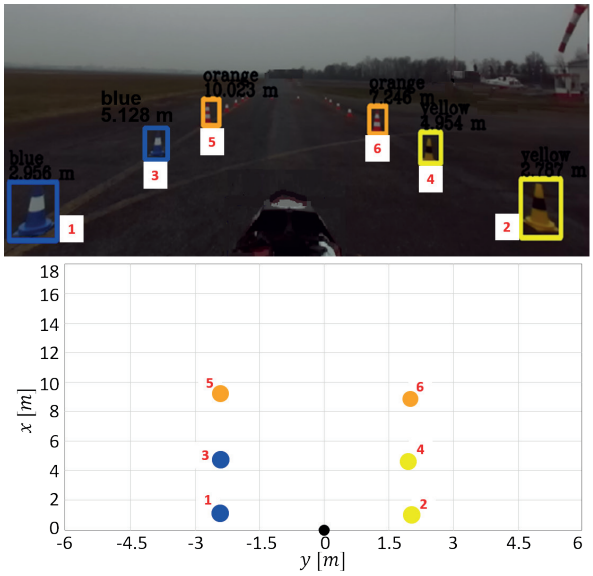

Figure 13 *Stereocamera-based detection results and resulting 2D local map (racetrack 2) while approaching the end of the track*

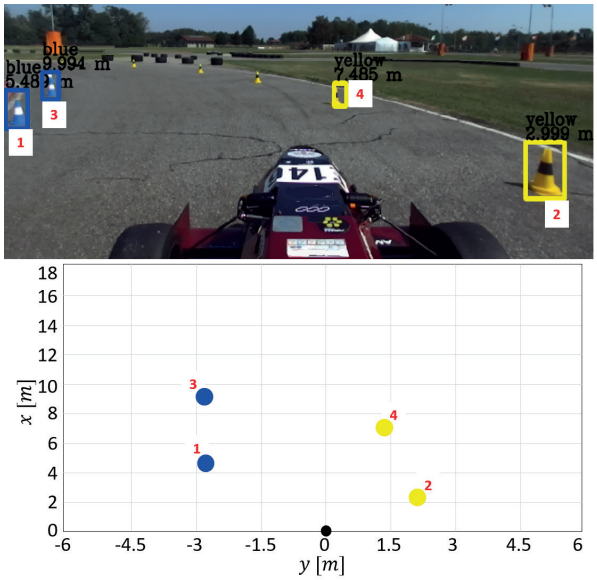

Figure 14 *Stereocamera-based detection results and resulting 2D local map during the cornering (racetrack 1, left curve)*

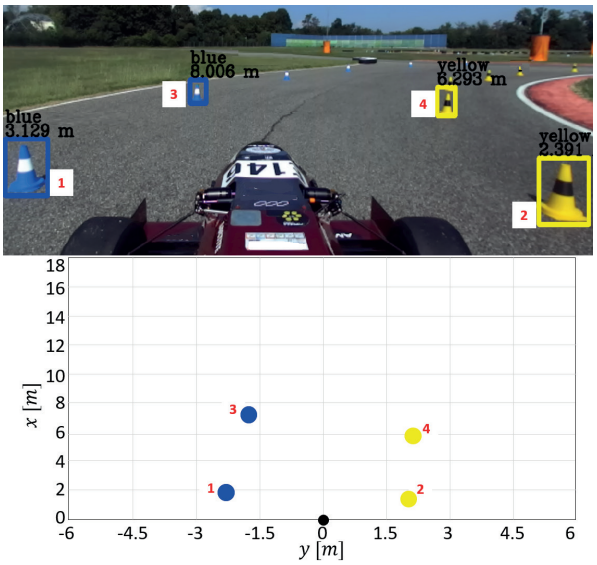

Figure 15 *Stereocamera-based detection results and resulting 2D local map during the cornering (racetrack 1, right curve)*

enabling its usage for autonomous driving purposes. The local map is actually created up to 20 m in the case of the LiDAR-based algorithm and up to 10 m using the stereocamera based method. The method is also robust with respect to possible outliers, being proven in a challenging environment, i.e. stripes on the ground and other types of objects in the environment that can affect the accuracy.

The distance estimation has been validated with respect to the ground truth provided by a roll-meter with a negligible error that is equal to few centimeters, as well as in the case of stereocamera-based algorithm.

The proposed method could also be implemented for any other kind of obstacles by means of properly changing the clustering parameters and by applying a proper transfer learning procedure for the SSD image classifier. This process is beyond the purpose of this paper since it is addressed to a peculiarly structured environment.

The algorithm is sufficiently redundant in the case of a failure in one of the sensors, thus avoiding common issues in the LiDAR-camera sensor fusion, when both measurements must always be provided to build a local map. Therefore, the local map can enable any further trajectory planning technique for autonomous driving.

4 Conclusion

Considering the recent innovations in the framework of the self-driving vehicles, a redundant multi-object detection algorithm has been presented. The proposed method exploits the combination of the LiDAR and stereocamera sensors, both to detect different obstacles and to create a local map by estimating the obstacles distance. The method can accurately filter out of the point-cloud the unnecessary points with a segmentation algorithm and identify the clusters relative to each obstacle in the structured environment. Moreover, it is very robust with respect to outliers in the vision-based perception algorithm, that is performed with an SSD architecture. The solution is presented as a reliable alternative to existing methods to guarantee redundancy on the local map, which is successfully created, frame by frame, up to 20 m in front of the vehicle. The performance of the method was evaluated experimentally during the real driving maneuvers, which have been performed by an all-wheel drive electric racing vehicle in a properly structured racetrack. The proposed approach is different from the sensor fusion at a sensor level, because the information is fused and synchronized in the local map building process via the two-dimensional over-imposition and synchronization of the detected obstacles. In fact, avoiding the sensor fusion at a sensor level can save computational costs, as no additional algorithms are deployed on the devoted control unit. According to the proposed perception method, creation of a local map, with respect to the vehicle position, is enabled for the deployment of further local trajectory planning algorithms. Consequently, an extensive validation stage of the method, considering several obstacles, is required before its deployment on commercial vehicles.

Acknowledgements

This research work was developed in the framework of the activities of the Interdepartmental Center for Automotive Research and Sustainable Mobility (CARS) at Politecnico di Torino (www.cars.polito.it).

References

[1] SILBERG, G., MANASSA, M., EVERHART, K., SUBRAMANIAN, D., CORLEY, M., FRASER, H., SINHA, V. *Self-driving cars: are we ready?* [online]. Netherlands: KPMG LLp, 2013. Available from: https://assets.kpmg/content/dam/kpmg/pdf/2013/10/self-driving-cars-are-we-ready.pdf

[2] FRAEDRICH, E., LENZ, B. Automated driving: individual and societal aspects. *Transportation Research Record: Journal of the Transportation Research Board* [online]. 2014, **2416**(1), p. 64-72. ISSN 0361-1981, eISSN 2169-4052. Available from: https://doi.org/10.3141/2416-08

[3] ZIEBINSKI, A., CUPEK, R., GRZECHCA, D., CHRUSZCZYK, L. Review of advanced driver assistance systems (ADAS). *AIP Conference Proceedings* [online]. 2017, **1906**(1), 120002. ISSN 0094-243X, eISSN 1551-7616. Available from: https://doi.org/10.1063/1.5012394

[4] KAUR, K., RAMPERSAD, G. Trust in driverless cars: Investigating key factors influencing the adoption of driverless cars. *Journal of Engineering and Technology Management* [online]. 2018, **48**, p. 87-96. ISSN 0923-4748. Available from: https://doi.org/10.1016/j.jengtecman.2018.04.006

[5] HORL, S., CIARI, F., AXHAUSEN, K. W. *Recent perspectives on the impact of autonomous vehicles* [online]. Working paper 10XX. Zurich: Institute for Transport Planning and Systems, 2016. Available from: https://doi.org/10.3929/ethz-b-000121359

[6] BONFITTO, A., FERACO, S., TONOLI, A., AMATI, N. Combined regression and classification artificial neural networks for sideslip angle estimation and road condition identification. *Vehicle System Dynamics* [online]. 2020, **58**(11), p. 1766-1787. ISSN 0042-3114, eISSN 1744-5159. Available from: https://doi.org/10.1080/00423114.2019.1645860

[7] BUEHLER, M., IAGNEMMA, K., SINGH, S. *The 2005 DARPA grand challenge: the great robot race* [online]. Vol. 36. Berlin Heidelberg: Springer, 2007. ISBN 978-3-540-73429-1, eISBN 978-3-540-73429-1. Available from: https://doi.org/10.1007/978-3-540-73429-1

[8] THRUN, S., MONTEMERLO, M., DAHLKAMP, H., STAVENS, D., ARON, A., DIEBEL, J., FONG, P., GALE, J., HALPENNY, M., HOFFMANN, G., LAU, K., OAKLEY, C., PALATUCCI, M., PRATT, V., STANG, P., STROHBAND, S., DUPONT, C., JENDROSSEK, L. - E., KOELEN, CH., MARKEY, CH., RUMMEL, C., VAN NIEKERK, J., JENSEN, E., ALESSANDRINI, P., BRADSKI, G., DAVIES, B., ETTINGER, S., KAEHLER, A., NEFIAN, A., MAHONEY, P. Stanley: The robot that won the DARPA Grand Challenge [online]. *Journal of Field Robotics*. 2006, **23**(9), p. 661-692. eISSN 1556-4967. Available from: https://doi.org/10.1002/rob.20147

[9] LITMAN, T. Autonomous vehicle implementation predictions: implications for transport planning. In: Transportation Research Board 94th Annual Meeting: proceedings. 2015. No. 15-3326.

[10] SINGH, S. *Critical reasons for crashes investigated in the national motor vehicle crash causation survey*. Washington, DC: National Center for Statistics and Analysis, 2015. No. DOT HS 812 115.

[11] YURTSEVER, E., LAMBERT, J., CARBALLO, A., TAKEDA, K. A survey of autonomous driving: common practices and emerging technologies. *IEEE Access* [online]. 2020, **8**, p. 58443-58469. eISSN 2169-3536. Available from: https://doi.org/10.1109/ACCESS.2020.2983149

[12] PENDLETON, S. D., ANDERSEN, H., DU, X., SHEN, X., MEGHJANI, M., ENG, Y. H., RUS, D., ANG, M. H. Perception, planning, control and coordination for autonomous vehicles. *Machines* [online]. 2017, **5**(1), 6. eISSN 2075-1702. Available from: https://doi.org/10.3390/machines5010006

[13] MARTI, E., DE MIGUEL, M. A., GARCIA, F., PEREZ, J. A review of sensor technologies for perception in automated driving. *IEEE Intelligent Transportation Systems Magazine* [online]. 2019, **11**(4), p. 94-108. ISSN 1939-1390, eISSN 1941-1197. Available from: https://doi.org/10.1109/MITS.2019.2907630

[14] KEMSARAM, N., DAS, A., DUBBELMAN, G. A stereo perception framework for autonomous vehicles. In: 2020 IEEE 91st Vehicular Technology Conference VTC2020-Spring: proceedings [online]. IEEE. 2020. eISBN 978-1-7281-5207-3, p. 1-6. Available from: https://doi.org/10.1109/VTC2020-Spring48590.2020.9128899

[15] FERACO, S., BONFITTO, A., KHAN, I., AMATI, N., TONOLI, A. Optimal trajectory generation using an improved probabilistic road map algorithm for autonomous driving. In: 22nd International Design Engineering Technical Conferences and Computers and Information in Engineering Conference: proceedings [online]. American Society of Mechanical Engineers. Vol. 83938. 2020. ISBN 978-0-7918-8393-8, V004T04A006. Available from: https://doi.org/10.1115/DETC2020-22311

[16] FERACO, S., BONFITTO, A., AMATI, N., & TONOLI, A. A LIDAR-based clustering technique for obstacles and lane boundaries detection in assisted and autonomous driving. In: 22nd International Design Engineering Technical Conferences and Computers and Information in Engineering Conference: proceedings [online]. American Society of Mechanical Engineers. Vol. 83938. 2020. ISBN 978-0-7918-8393-8, V004T04A007. Available from: https://doi.org/10.1115/DETC2020-22339

[17] WANG, M., LIU, W. - Q., LU, Y. - H., ZHAO, X. - S., SONG, B. - CH., ZHANG, Y. - J., WANG, Y. - P., LIAN, C. - H., CHEN, J., CHENG, Y., LIU, J. - G., WEI, Q. - N. Study on the measurement of the atmospheric extinction of fog and rain by forward-scattering near infrared spectroscopy. *Spectroscopy and Spectral Analysis*. 2008, **28**(8), p. 1776-1780.

[18] PHILLIPS, T. G., GUENTHER, N., MCAREE, P. R. When the dust settles: the four behaviors of LIDAR in the presence of fine airborne particulates. *Journal of Field Robotics* [online]. 2017, **34**(5), p. 985-1009. eISSN 1556-4967. Available from: https://doi.org/10.1002/rob.21701

[19] GOSALA, N., BUHLER; A., PRAJAPAT, M., EHMKE, C., GUPTA, M., SIVANESAN, R., GAWEL, A., PFEIFFER, M., BURKI, M., SA, I., DUBE, R., SIEGWART, R. Redundant perception and state estimation for reliable autonomous racing. In: 2019 International Conference on Robotics and Automation ICRA 2019: proceedings [online]. IEEE. 2019. Available from: https://doi.org/10.1109/ICRA.2019.8794155

[20] KUMAR, G. A., LEE, J. H., HWANG, J., PARK, J., YOUN, S. H., KWON, S. LiDAR and camera fusion approach for object distance estimation in self-driving vehicles. *Symmetry* [online]. 2020, **12**(2), 324. eISSN 2073-8994. Available from: https://doi.org/10.3390/sym12020324

[21] ROSIQUE, F., NAVARRO, P. J., FERNANDEZ, C., PADILLA, A. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors* [online]. 2019, **19**(3), 648. eISSN 1424-8220. Available from: https://doi.org/10.3390/s19030648

[22] LEE, K. B., HAN, M. H. Lane-following method for high speed autonomous vehicles. *International Journal of Automotive Technology* [online]. 2008, **9**(5), p. 607-613. ISSN 1229-9138. eISSN 1976-3832. Available from: https://doi.org/10.1007/s12239-008-0072-z

[23] CHU, K., KIM, J., JO, K., SUNWOO, M. Real-time path planning of autonomous vehicles for unstructured road navigation. *International Journal of Automotive Technology* [online]. 2015, **16**(4), p. 653-668. ISSN 1229-9138. eISSN 1976-3832. Available from: https://doi.org/10.1007/s12239-015-0067-5

[24] HU, J., XIONG, S., ZHA, J., FU, C. Lane detection and trajectory tracking control of autonomous vehicle based on model predictive control. *International Journal of Automotive Technology* [online]. 2020, **21**(2), p. 285-295. ISSN 1229-9138. eISSN 1976-3832. Available from: https://doi.org/10.1007/s12239-020-0027-6

[25] KHAN, I., FERACO, S., BONFITTO, A., AMATI, N. A model predictive control strategy for lateral and longitudinal dynamics in autonomous driving. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference ASME 2020: proceedings. American Society of Mechanical Engineers Digital Collection. 2020.

[26] HIMMELSBACH, M., HUNDELSHAUSEN, F. V., WUENSCHE, H. J. Fast segmentation of 3D point clouds for ground vehicles. In: 2010 IEEE Intelligent Vehicles Symposium: proceedings. IEEE. 2010. ISBN 978-1-4244-7866-8, p. 560-565.

[27] ZERMAS, D., IZZAT I., PAPANIKOLOPOULOS, N. Fast segmentation of 3D point clouds: a paradigm on LiDAR data for autonomous vehicle applications. In: 2017 IEEE International Conference on Robotics and Automation ICRA 2017: proceedings [online]. IEEE, 2017. eISBN 9781509046331, ISSN 1050-4729. Available from: https://doi.org/10.1109/ICRA.2017.7989591

[28] NYGREN, P., JASINSKI, M. *A comparative study of segmentation and classification methods for 3D point clouds* [online]. MS thesis. Gothenburg, Sweden: Chalmers University of Technology, University of Gothenburg, 2016. Available from: https://hdl.handle.net/20.500.12380/238602

[29] ZHOU, Y., TUZEL, O. Voxelnet: End-to-end learning for point cloud based 3D object detection. In: IEEE Conference on Computer Vision and Pattern Recognition CVPR 2018: proceedings [online]. 2018. p. 4490-4499. Available from: https://doi.org/10.1109/CVPR.2018.00472

[30] LIU, W., SUN, J., LI, W., HU, T., WANG, P. Deep learning on point clouds and its application: a survey. *Sensors* [online], 2019, **19**(19), 4188. eISSN 1424-8220. Available from: https://doi.org/10.3390/s19194188

[31] GRILLI, E., MENNA, F., REMONDINO, F. A review of point clouds segmentation and classification algorithms. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* [online]. 2017, **XLII**, p. 339-344. eISSN 2194-9034. Available from: https://doi.org/10.5194/isprs-archives-XLII-2-W3-339-2017

[32] GUO, Y., WANG, H., HU, Q., LIU, H., LIU, L., BENNAMOUN, M. Deep learning for 3D point clouds: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. 2020, early access. ISSN 0162-8828, eISSN 1939-3539. Available from: https://doi.org/10.1109/TPAMI.2020.3005434

[33] LIU, W., ANGUELOV, D., ERHAN, D., SZEGEDY, C., REED, S., FU, C. Y., BERG, A. C. SSD: Single shot multibox detector. In: European Conference on Computer Vision: proceedings [online]. Springer, Cham. 2016. ISBN 978-3-319-46447-3, eISBN 978-3-319-46448-0, p. 21-37. Available from: https://doi.org/10.1007/978-3-319-46448-0_2

[34] ZHAO, Z. Q., ZHENG, P., XU, S. T., WU, X. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems* [online]. 2019, **30**(11), p. 3212-3232. ISSN 2162-237X, eISSN 2162-2388. Available from: https://doi.org/10.1109/TNNLS.2018.2876865

[35] JIAO, L., ZHANG, F., LIU, F., YANG, S., LI, L., FENG, Z., QU, R. A survey of deep learning-based object detection. *IEEE Access* [online]. 2019, **7**, p. 128837-128868. eISSN 2169-3536. Available from: https://doi.org/10.1109/ACCESS.2019.2939201

[36] KABZAN, J., VALLS, M. I., REIJGWART, V. J., HENDRIKX, H. F., EHMKE, C., PRAJAPAT, M., BUHLER, A., GOSALA, N., GUPTA, M., SIVANESAN, R., DHALL, A., CHISARI, E., KARNCHANACHARI, N., BRITS, S., DANGEL, M., SA, I., DUBE, R., GAWEL, A., PFEIFFER, M., LINIGER, A., LYGEROS, J., SIEGWART, R. AMZ driverless: The full autonomous racing system. *Journal of Field Robotics* [online]. 2020, **37**(7), p. 1267-1294. eISSN 1556-4967. Available from: https://doi.org/10.1002/rob.21977

[37] GLENNIE, C. L., KUSARI, A., FACCHIN, A. Calibration and stability analysis of the VLP-16 laser scanner. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* [online]. 2016, **XL**, p. 55-60. eISSN 2194-9034. Available from: https://doi.org/10.5194/isprs-archives-XL-3-W4-55-2016

[38] ORTIZ, L. E., CABRERA, E. V., GONÇALVES, L. M. Depth data error modeling of the ZED 3D vision sensor from stereolabs. *ELCVIA: Electronic Letters on Computer Vision And Image Analysis* [online]. 2018, **17**(1), p. 1-15. eISSN 1577-5097. Available from: https://doi.org/10.5565/rev/elcvia.1084

[39] DITTY, M., KARANDIKAR, A., REED, D. Nvidia's xavier SoC. I: Hot Chips: A Symposium on High Performance Chips: proceedings. 2018.

[40] FSG competition handbook 2019 - Formula Student Germany [online]. 2019. Available from: https://www.formulastudent.de/fileadmin/user_upload/all/2019/rules/FSG19_Competition_Handbook_v1.0.pdf

[41] WANG, S., WU, T., VOROBEYCHIK, Y. Towards robust sensor fusion in visual perception [online]. 2020. Available from: arXiv:2006.13192

[42] URMSON, C., ANHALT, J., BAGNELL, D., BAKER, C., BITTNER, R., CLARK, M. N., DOLAN, J., DUGGINS, D., GALATALI, T., GEYER, CH., GITTLEMAN, M., HARBAUGH, S., HEBERT, M., HOWARD, T. M., KOLSKI, S., KELLY, A., LIKHACHEV, M., MCNAUGHTON, M., MILLER, N., PETERSON, K., PILNICK, B., RAJKUMAR, R., RYBSKI, P., SALESKY, B., SEO, Y. - W., SINGH, S., SNIDER, J., STENTZ, A., WHITTAKER, W. "R.", WOLKOWICKI, Z., ZIGLAR, J., BAE, H., BROWN, T., DEMITRISH, D., LITKOUHI, B., NICKOLAOU, J., SADEKAR, V., ZHANG, W., STRUBLE, J., TAYLOR, M., DARMS, M., FERGUSON, D. Autonomous driving in urban environments: boss and the urban challenge. *Journal of Field Robotics* [online]. 2008, **25**(8), p. 425-466. eISSN 1556-4967. Available from: https://doi.org/10.1002/rob.20255

[43] KAMMEL, S., ZIEGLER, J., PITZER, B., WERLING, M., GINDELE, T., JAGZENT, D., SCHRODER, J., THUY, M., GOEBL, M., VON HUNDELSHAUSEN, F., PINK, O., FRESE, CH., STILLER, CH. Team AnnieWAY's autonomous system for the 2007 DARPA urban challenge. *Journal of Field Robotics* [online]. 2008, **25**(9), p. 615-639. eISSN 1556-4967. Available from: https://doi.org/10.1002/rob.20252

[44] MONTEMERLO, M., BECKER, J., BHAT, S., DAHLKAMP, H., DOLGOV, D., ETTINGER, S., HAEHNEL, D., HILDEN, T., HOFFMANN, G., HUHNKE, B., JOHNSTON, D., KLUMPP, S., LANGER, D., LEVANDOWSKI, A., LEVINSON, J., MARCIL, J., ORENSTEIN, D., PAEFGEN, J., PENNY, I., PETROVSKAYA, A., PFLUEGER, M., STANEK, G., STAVENS, D., VOGT, A., THRUN, S. Junior: the Stanford entry in the urban challenge. *Journal of Field Robotics* [online]. 2008, **25**(9), p. 569-597. eISSN 1556-4967. Available from: https://doi.org/10.1002/rob.20258

[45] MOOSMANN, F., PINK, O., STILLER, C. Segmentation of 3D LiDAR data in non-flat urban environments using a local convexity criterion. In: IEEE Intelligent Vehicles Symposium IV 09: proceedings [online]. 2009. ISSN 1931-0587. Available from: https://doi.org/10.1109/IVS.2009.5164280

[46] KLASING, K., WOLLHERR, D., BUSS, M. Realtime segmentation of range data using continuous nearest neighbors. In: 2009 IEEE International Conference on Robotics and Automation ICRA: proceedings 2009.

[47] KLASING, K., WOLLHERR, D., BUSS, M. A clustering method for efficient segmentation of 3D laser data. In: IEEE International Conference on Robotics and Automation ICRA 2008: proceedings. 2008. p. 4043-4048.

[48] STEINHAUSER, D., RUEPP, O., BURSCHKA, D. Motion segmentation and scene classification from 3D LiDAR data. In: IEEE Intelligent Vehicles Symposium: proceedings. 2008. p. 398-403.

[49] ANGUELOV, D., TASKAR, B., CHATALBASHEV, V., KOLLER, D., GUPTA, D., HEITZ, G., NG, A. Discriminative learning of Markov random fields for segmentation of 3D scan data. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'05: proceedings. Vol. 2. 2005. p. 169-176.

[50] RUSU, R. B. Semantic 3D object maps for everyday manipulation in human living environments. *KI-Kunstliche Intelligenz* [online]. 2010, **24**(4), p. 345-348. ISSN 0933-1875, eISSN 1610-1987. Available from: https://doi.org/10.1007/s13218-010-0059-6

[51] RUSU, R. B., COUSINS, S. 3D is here: Point cloud library (PCL). In: 2011 IEEE International Conference on Robotics and Automation: proceedings. 2011. p. 1-4.

[52] YAN, Z., DUCKETT, T., BELLOTTO, N. Online learning for 3D LiDAR-based human detection: experimental analysis of point cloud clustering and classification methods. *Autonomous Robots* [online]. 2020, **44**(2), p. 147-164. ISSN 0929-5593, eISSN 1573-7527. Available from: https://doi.org/10.1007/s10514-019-09883-y

[53] BENTLEY, J. L. Multidimensional binary search trees used for associative searching. *Communications of the ACM* [online]. 1975, **18**(9), p. 509-517. ISSN 0001-0782, eISSN 1557-7317. Available from: https://doi.org/10.1145/361002.361007

[54] MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. In: 5-th Berkeley Symposium on Mathematical Statistics and Probability: proceedings. Berkeley, University of California Press. Vol. 1. 1967. p. 281- 297.

[55] HOWARD, A. G., ZHU, M., CHEN, B., KALENICHENKO, D., WANG, W., WEYAND, T., ANDREETTO, M., ADAM, H. Mobile nets: efficient convolutional neural networks for mobile vision applications [online]. 2017. Available from: http://arxiv.org/abs/1704.04861

[56] GALVEZ, R. L., BANDALA, A. A., DADIOS, E. P., VICERRA, R. R. P., MANINGO, J. M. Z. Object detection using convolutional neural networks. In: IEEE Region 10 Conference TENCON 2018: proceedings. 2018. p. 2023-2027.

[57] ZHUANG, F., QI, Z., DUAN, K., XI, D., ZHU, Y., ZHU, H., XIONG, H., HE, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE* [online]. 2020, **109**(1), p. 43-76. Available from: https://doi.org/10.1109/JPROC.2020.3004555

[58] R. SZELISKY, Computer vision: algorithms and applications [online]. London: Springer Link, 2012. ISBN 978-1-84882-934-3, eISBN 978-1-84882-935-0. Available from: https://doi.org/10.1007/978-1-84882-935-0

[59] ADI, K., WIDODO, C. E. Distance measurement with a stereo camera. *International Journal of Innovative Research in Advanced Engineering* [online]. 2017, **4**(11), p. 24-27. ISSN 2349-2163. Available from: \https://doi.org/10.26562/IJIRAE.2017.NVAE10087

[60] ZAARANE, A., SLIMANI, I., AL OKAISHI, W., ATOUF, I., HAMDOUN, A. Distance measurement system for autonomous vehicles using stereo camera. *Array* [online]. 2020, **5**, 100016. ISSN 2590-0056. Available from: https://doi.org/10.1016/j.array.2020.100016