

pixel coordinates in the image space of the AR interface are calculated through projection and a color is assigned indicating the object class. Interfaces that can be used for in-vehicle visualization include AR headset, Head-Up Display (HUD) [17], [18] or even the car's windshield with transparent display.

The contributions of the proposed approach can be summarized as follows.

- Development of an obstacle detection module that takes into account the extraction of saliency maps from point clouds.
- Generation of data for randomized multi-ego connected vehicle in cooperative driving scenarios.
- Creation of realistic synthetic data of potholes that can be entered in the town maps of the CARLA simulator for the design of lifelike driving situations.
- AR visualization for point cloud projection registered on the scene images.
- Development of public and open access libraries with code for the aforementioned components^{1 2 3 4}

The rest of this paper is organized as follows. First we present previous works in related domains in Section II and then describe in detail the proposed methodology in Sections III and IV. Section V follows with some experimental results in comparison with other state-of-the-art methods, while Section VI draws the conclusions and directions for future work.

II. PREVIOUS WORK

In the following we provide an overview of methodologies tackling the main challenges of the presented approach on (i) obstacle detection, (ii) cooperative driving and (iii) AR infotainment systems.

1) *Obstacle Detection*: A major element that adds unpredictability in path planning for self-driving cars are obstacles in the road. Obstacles can appear in the form of objects beyond the surface of the road, or cracks and holes in paved areas. There has been major work on obstacle detection, ranging from real-time implementations [19], to offline schemes that act as automated informants to the authorities responsible for maintenance [20], or as efficient unsupervised techniques for pothole detection [21]. Most of the existing works implement a broad spectrum of computer vision and/or machine learning techniques to analyze imaging information [19]. The methods differ mainly on the utilized features and classifiers for obstacle representation and recognition. In respect to performance, a direct comparison of methods is not feasible because most works are evaluated on their own (simulated) data. In fact, there is lack or restricted access to a common benchmark dataset with potholes and obstacles, that can be used for comparison.

Waga *et al.* [22] used superpixel segmentation to partition the image into superpixels based on the entropy rate, and then applied Support Vector Machines (SVM) to estimate the probability of each superpixel being the part of some

object based on textural features (namely histogram of oriented gradients, co-occurrence matrix, intensity histogram and mean intensity). For final object label inference, merging of the superpixels is performed using conditional random fields to account for neighborhood similarity. The drawbacks of this method are its dependency only on texture information and more specifically the inability to distinguish between a shadow and a hole easily, leading to potential false positives.

Another image-based method that takes advantage of the texture characteristics of potholes is the work of Kanza *et al.* [23]. Here, the histogram of oriented gradients (HOG) is extracted from the grayscale image and coupled with a Naive Bayes classifier. If the probability calculated by the classifier is high enough, then the pothole localization is performed using graph-based segmentation and normalized cuts. This method presents very encouraging results for the examined dataset that simulates a variety of cases and conditions, including changes in illumination and potholes filled with water.

Yifan *et al.* [24] take a different direction and use Unmanned Aerial Vehicle (UAV) for pothole detection in the suburb of Shihenzi City. The aerial images are segmented and the segmented parts are used to extract features, including the mean, standard deviation, area, length/width ratio, elliptic fit, roundness, contrast, dissimilarity, homogeneity and correlation. Segmentation is performed using a multiresolution segmentation algorithm that is integrated into the eCognition Developer software (a development environment for object-based analysis of geospatial data). For classification, the SVM, Artificial Neural Networks (ANN), and Random Forest (RF) classifiers are compared, each with different combinations of features while also taking the computational time into consideration. The authors conclude that spatial features (texture and geometry) coupled with RF are the most effective, although this method is very sensitive to UAV image resolution, weather and lighting conditions.

Other methods focus on road cracks detection from high resolution cameras on smartphones. Since such data are more easily available, those methods can bypass the extraction of hand-crafted features and utilize deep architectures, such as convolutional neural networks [25], [26], [27], [28]. However, in the case of dense traffic situations and poor lighting conditions, techniques utilizing images from smartphone camera are less effective.

In contrast to computer vision techniques which exploit texture information from images, 3D point cloud processing techniques exploit the object's geometrical properties [15], [29], [30]. Bosurgi *et al.* [30] identify potholes in road sections by estimating area, perimeter and depth information from 3D data of pavement surfaces. Chen *et al.* [15] propose a framework for obstacle detection using the pitch and rotation angles of a LiDAR sensor to create a 2D image-like plane where the unordered set of points (from the point cloud) are projected. From this "LiDAR-imagery" a 2D histogram is extracted and used to find the road plane. If an adequate part of the road, in front of the vehicle is flat, those points form a straight line in the histogram representation, and anything above the line can be classified as a positive obstacle (points higher than the road plane), while points below the line as a

¹<https://github.com/Stagakis/saliency-from-pointcloud>

²<https://github.com/Stagakis/carla-data-generation>

³<https://github.com/Stagakis/roadpatch-with-pothole-generator>

⁴<https://github.com/Stagakis/carlapclprocessing>