threshold of 15% reshape in the area in either direction to avoid frequent unnecessary updates, while also retaining the required precision in representation. Similarly, the algorithm checks for significant changes in the bounding box dimensions. A flowchart showcasing the information update and communication pipeline between two vehicles is shown in Fig. 13

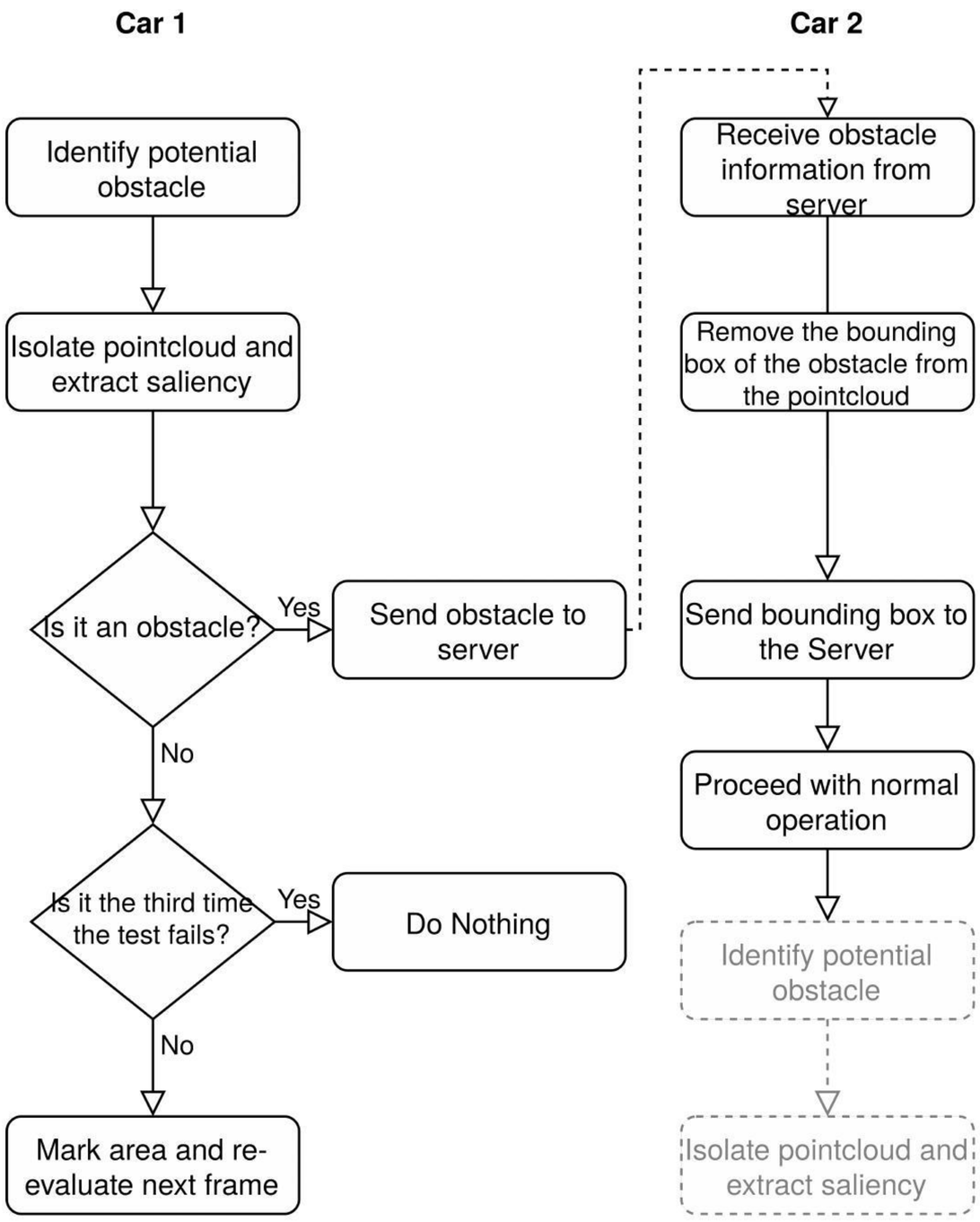

Fig. 13. Flowchart of communicative vehicles for obstacle sharing.

V. Experimental Analysis

In this section, we will present and discuss in detail the experimental analysis and will evaluate our proposed framework.

A. Experimental Setup, Datasets and Metrics

The experiments were carried out on an Intel Core i7-4790HQ CPU @ 3.60GHz PC with 16 GB of RAM. The core algorithms are written in Matlab and C++. The evaluation of the methodology was performed using (i) synthetic dataset of potholes that we have created and (ii) 3D point cloud potholes from real datasets with known models (used as ground truth) which have been evaluated by other methods too [63], [64], [65], [66].

The pothole detection algorithms are compared in terms of the pixel-level (for image-based methods) and point-level (for point clouds) $precision = [TP/(TP + FP)]$, $recall = [TP/(TP+FN)]$, $accuracy = [(TP+TN)/(TP+TN + FP + FN)]$ and $F - score = 2 \cdot [(precision \cdot recall)/(precision + recall)]$, where TP, FP, TN, FN, represent the number of True-Positive, False-Positive, True-Negative and False-Negative pixels, respectively. The positive class includes all vertices belonging to the pothole (P) and the negative class all vertices belonging to the road (R). The performance metrics can also be expressed as shown in Table I, where *Real Pothole* (RP) represents the recall or in other words the percentage of vertices correctly annotated as pothole, *Real Road* (RR) represents the percentage of vertices correctly annotated as road, *Not real Pothole* (NP) represents the percentage of vertices wrongly annotated as pothole and *Not real Road* (NR) represents the percentage of vertices wrongly annotated as road.

TABLE I
Evaluation metrics for pothole detection (in percentage %)

(×100%)	Annotated as Pothole	Annotated as Road
Actual Pothole	RP = $\frac{TP}{P}$	NR = 1 - RP
Actual Road	NP = 1 - RR	RR = $\frac{TN}{R}$

B. Results

For the evaluation of our method, two public available datasets [63], [64] were utilized providing point clouds of real potholes. Fig. 14 visualizes results of our pothole detection method for the dataset created by real potholes [63] under different density resolutions (14 (a)-(d)). Points in red represent the vertices belonging to the pothole, while points in blue represent vertices belonging to the road, both for the ground truth and the estimated point clouds. Two dense models (14 (a)) are utilized as presented in rows 1-3 and 4-6, respectively. To investigate the performance of our approach in more realistic conditions, we increasingly downsampled the original point cloud (14 (b)-(c)) to evaluate the robustness of detection of our algorithm. The corresponding number of vertices for the two models (original and downsampled) are shown above each model, respectively. The heatmap (rows 2 and 5) illustrates the geometric and spectral saliency per vertex (as estimated from Eq. 8). Higher salient values are depicted with deep red color while lower salient values with deep blue.

Due to the sensitive nature of the specific application involving safety of drivers (via information visualization for situational awareness), we prefer our algorithm to provide a small percentage of NR than having even a small value of NP (please refer to Table I). To wrongly identify as a pothole a small area of the road around an actual pothole is not as critical in our application as the opposite, namely to fail to present or partially present a potentially dangerous object (e.g., pothole, ramp).

The detailed results with all evaluation metrics are shown in Table II for each of the thirteen 3D models of the point cloud dataset, and under different point cloud density resolutions. The results of this table show that our method is robust even for very low point cloud density. This is an important observation, since the output of the LiDAR device has a low density resolution pattern.

Fig. 15 visualizes some examples of the pothole detection algorithm applied in an other dataset [64]. The first column of