Camera and LiDAR analysis for 3D object detection in foggy weather conditions

Nguyen Anh Minh MAI
STI research team, Cerema
and
IRIT, Université de Toulouse, UPS
Toulouse, France
0000-0001-9672-8909

Pierre DUTHON
STI research team, Cerema
Clermont-Ferrand, France
0000-0002-6705-1131

Pascal Housam SALMANE
STI research team, Cerema
Toulouse, France
0000-0002-0919-7482

Louahdi KHOUDOUR
STI research team, Cerema
Toulouse, France
0000-0002-5947-4302

Alain CROUZIL
IRIT, Université de Toulouse, UPS
Toulouse, France
0000-0001-7040-2978

Sergio A. VELASTIN
School of EECS
Queen Mary University of London, UK
and
Dept. of Computer Science
University Carlos III, Madrid, Spain
0000-0001-6775-7137

***Abstract*—Today, the popularity of self-driving cars is growing at an exponential rate and is starting to creep onto the roads of developing countries. For autonomous vehicles to function, one of the essential features that needs to be developed is the ability to perceive their surroundings. To do this, sensors such as cameras, LiDAR, or radar are integrated to collect raw data. The objective of this paper is to evaluate a fusion solution of cameras and LiDARs (4 and 64 beams) for 3D object detection in foggy weather conditions. The data from the two input sensors are fused and an analysis of the contribution of each sensor on its own is then performed. In our analysis, we calculate average precision using the popular KITTI dataset, on which we have applied different intensities of fog (on a dataset we have called Multifog KITTI). The main results observed are as follows. Performances with stereo camera and 4 or 64 beams LiDAR are high (90.15%, 89.26%). Performance of the 4 beams LiDAR alone decreases sharply in foggy weather conditions (13.43%). Performance when using only a camera-based model remains quite high (89.36%). In conclusion, stereo cameras on their own are capable of detecting 3D objects in foggy weather with high accuracy and their performance slightly improves when used in conjunction with LIDAR sensors.**

***Index Terms*—self-driving car, 3D object detection, foggy weather condition, sensor fusion**

I. Introduction

Nowadays, self-driving cars are becoming more and more popular in developed countries. These cars have to make accurate predictions such as lane detection, traffic sign detection and obstacle detection. Due to the increasing importance and attractiveness of autonomous vehicles, a number of self-driving car companies have emerged, and almost every traditional automaker has invested in this segment to some extent.

Much has been invested in researching hardware and software for self-driving cars, making it an extremely hot research topic. To better perceive the environment, self-driving cars are equipped with sensors such as cameras, radars and LiDARs. These sensors are used to collect data about the environment. Then, the data is fed into predictive models such as semantic classification, segmentation, and object detection to obtain the information needed for vehicle navigation. In this way, the vehicle can detect lanes, obstacles, and traffic signs to make decisions such as wheel speed or steering wheel angle. These models must perform with very high accuracy while maintaining high safety standards. Therefore, the accuracy of predictive models plays an important role in the operation of self-driving cars.

In previous studies on environment perception, either a camera-based [3], [6], [18], [19], [21] or LIDAR-based [5], [11]–[15] detection algorithms were developed. The data was collected under ideal conditions (daylight, without bad weather). Such studies showed that LiDAR-based models performed better than camera-based ones. This is because LiDAR is an active sensor able to provide a very accurate estimation of the distance from a self-driving car to an obstacle, while a camera does not, especially at long distances. Meanwhile, methods that combine both camera and LiDAR have not produced the desired results that could take advantage of the strengths of both systems [26]–[28]. Since 3D object detection methods now provide stable results on data under normal weather conditions , more attention needs to be paid to study how these models can be used in extreme weather conditions such as rain, snow, or fog [1], [10]. Furthermore, we do not know which model (camera-based, LiDAR-based, or fusion-based method) works best under these conditions. There are many studies showing that data collected by LiDAR, and cameras are significantly distorted in extreme weather conditions, e.g. [1], [31], [32]. However, there are not many

978-1-6654-6694-3/22/$31.00

studies that show the impact of these distorted data on the performance of the predictive model for 3D object detection. The main objective of this study is to analyze the contribution of the two sensors, i.e. camera and LiDAR, to 3D object detection in foggy weather conditions. The contribution of this paper is threefold:

- We divide and adapt the SLS-Fusion neural network [2] to take into account stereo camera or LiDAR separately. This leads to two different subneural networks.
- We study the performance of each subneural networks in foggy weather conditions in comparison with SLS-Fusion.
- We then analyze the performance of the 3D object detection models (for stereo camera and LiDAR) according to six levels of fog applied to KITTI dataset.

After this introduction, in Section II there is a brief summary of the state of the art in 3D obstacle detection by camera and LiDAR, the two sensors being taken together or separately. Section III describes the datasets used for the experiments. Section IV explains the methodology chosen for 3D object detection, the nature of the neural network used, and the methodology for using the network when considering stereo camera and LiDAR both jointly and separately, to analyze their influence. Finally, Section V presents the results on the performance of 3D object detection by camera and LiDAR. A conclusion and short-term perspectives (Section VII) ends the article.

II. Related work

This section places the proposed work in context. We distinguish between three main relevant topics: camera-based 3D object detection in Section II-A, LiDAR-based 3D object detection in Section II-B and fusion-based 3D object detection in Section II-C.

A. Camera-based 3D object detection

With the success of deep learning-based methods for camera-based classification and object recognition problems, this has been followed up and extended to 3D object detection. However, methods based only on camera sensors [3], [6], [18], [19], [21] have difficulty determining precise depth, the core element of the 3D vision problem. Deep MANTA [19] is an early work on monocular 3D object detection. It finds the optimal position by matching 2D and 3D keypoints. To recover 3D geometry from 2D data, a 3D template is selected from a template database based on the predicted similarity between 3D templates. However, this requires maintaining a huge database of 3D models and discards valuable information that can be regressed directly from the image. By using several geometric properties, Deep3DBox [20] utilizes the property that the perspective projection of 3D corners should touch at least one of the two 2D bounding boxes. In the case of deterministic geometry constraints, any errors in 2D object detection are locked into the 3D estimation. In MLF [21], a multi-stage fusion algorithm is used to combine image features with pseudo-LiDAR data. In that work, there is no optimization process, only depth is combined with an RGB image, which leads to suboptimal results. Since depth information is essential for 3D detection, pseudo point cloud using monocular or binocular data has been proposed [6]. That paper highlights the inefficiency of current methods for 3D object detection based on RGB-D images. A pseudo-3D point cloud in LiDAR coordinate system is generated by first predicting the depth map and then back-projecting. According to [6], image-based depth maps can be converted into a pseudo-LiDAR representation.

B. LiDAR-based 3D object detection

LiDAR does not provide information about the scene as clearly as an image, but the point clouds obtained from LiDAR are accurate estimations of the location of objects in space. This is an important contribution that helps make LiDAR-based methods superior to camera-based methods for 3D object detection. They are usually divided into 3 types: multi-view-based methods [11], voxel-based methods [12], [13], [15] and point-based methods [5], [14]. In [11], a new single-stage detector is proposed that performs 2D convolution on the Bird's-Eye-View (BEV). The objects to be detected do not overlap in the BEV representation, which maintains the physical scale. In addition, the BEV has a low computational cost and is one of the fastest detectors. In [12], sparse convolution operations are proposed to reduce memory consumption and increase computational speed. In sparse convolution, convolution is performed only over non-empty voxels and not over all voxels, as in traditional 3D convolution. With many optimizations to adjust the representation differences, Faster RCNN [16] and Mask RCNN [17] have been extended to point cloud representations and are proposed in PointRCNN [5]. The main advantage is to generate high quality 3D bounding box proposals from a point cloud.

C. Fusion-based 3D object detection

Since both camera and LiDAR data have their own advantages and disadvantages, various methods [26]–[28] have been explored to incorporate both data into the same model, but overall no significant superiority has been shown. The Frustum PointNets method [26] uses a standard 2D CNN object detector to extract 2D regions, and then converts the coordinates of the 2D regions into 3D space to create Frustum proposals. A PointNet-like block is used to segment each point within the frustum and obtain points of interest. Sophisticated 2D recognition methods are used to obtain prior knowledge, reducing the 3D search space and paving the way for its successors. This method depends heavily on the accuracy of 2D detectors, which is the main drawback of such a cascade approach. In PointPainting [28], semantic segmentation information from images is used to consolidate point clouds. To be precise, PointPainting first applies semantic information to each pixel to classify it. Then, a segmentation result, which is actually a compact outline of the image features, is "painted" onto the LiDAR points by projecting the LiDAR points directly onto the segmentation mask. Finally, a 3D detector can be

used with LiDAR for applications such as localization and classification. PointPainting uses segmentation values instead of RGB attributes to enhance existing LiDAR-based networks. Pseudo-LiDAR++ [3] uses Pseudo-LiDAR [6] as the basis for developing a 3D detection architecture. To rectify the depth estimation, Pseudo-LiDAR++ uses sparse 3D measurements (synthetic 4 beams LiDAR). The architecture is independent of any kind of LiDAR. However, this camera and LiDAR fusion needs to be studied further for harsh weather conditions like fog.

III. Datasets used

Detecting objects in foggy weather conditions is an important task for a self-driving car. For this reason, datasets collected from sensors such as LiDAR and cameras must be effectively fused into the model to improve the perception system of autonomous vehicles. To achieve better 3D object detection via depth maps, we use a new approach called Sparse LiDAR and Stereo Fusion (SLS-Fusion) [2] to fuse the data from stereo camera and LiDAR sensors.

To enable the operation of the SLS-Fusion system, the synthetic Scene Flow dataset [9] is first used to train the neural network model for depth estimation. This dataset collection contains more than 35,000 stereo images with a size of 960×540 and a large ground truth variety for optical flow and disparity. After depth prediction, the KITTI dataset [8] is used to train the 3D object detection part of the SLS-Fusion method. KITTI was chosen because it is one of the most common datasets for autonomous driving. It contains hours of real traffic scenarios recorded with both stereo and LiDAR sensors under normal weather conditions. A total of 7,481 training samples and 7,518 test samples are included. As detailed in [29], the training dataset is divided into two parts: training (3,712 samples) and validation (3,769 samples). On the other hand, there are few datasets recorded in extreme weather conditions. To address this problem, we decided to create a dataset, derived from KITTI, augmented with foggy weather conditions. To do this, we use a physics-based fog simulator [10] that converts normal weather data from KITTI to foggy weather data (this modified dataset is then called Multifog KITTI). In this dataset, the level of fog is characterised according to a visibility distance expressed in meters. Based on the results of the simulated foggy system, we divide foggy weather into six conditions or levels according to visibility (level 1: 20-29 m, level 2: 30-39 m, level 3: 40-49 m, level 4: 50-59 m, level 5: 60-69 m, and level 6: 70-79 m).

The detection results of the SLS-Fusion method were obtained by training with the Multifog KITTI data [1]. The performance of the 3D object detector can be analyzed by comparing the 3D predictor results with the annotated KITTI data results. However, after creating the new dataset by using the fog simulator, we found that many labeled objects (ground truths) that were visible without fog became invisible in fog, resulting in incorrect ground truths. For this reason, the ground truths were filtered. The bounding boxes, which have a greater distance to the sensor than the meteorological visibility, are removed and not taken into account in the evaluation. When we manually review the LiDAR and camera data, we see that this method does remove objects that are completely invisible.

IV. Methodology

This section presents the 3D object detection algorithm that we used in all experiments. The original model, which considered stereo camera and LiDAR simultaneously, was modified. Namely, a new representation was used for either the stereo camera or the LiDAR.

A. Initial model representation: SLS-Fusion

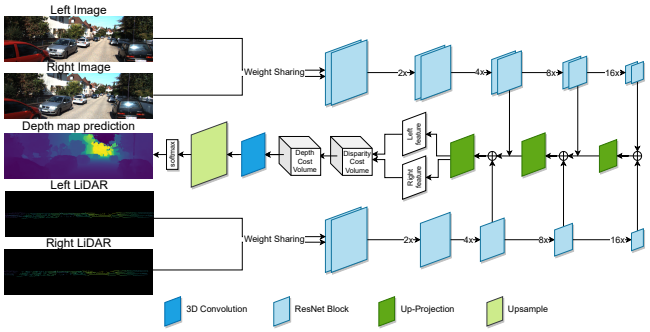

Fig. 1: Initial part of the depth estimation of the SLS-Fusion algorithm [2] with stereo camera and LiDAR as inputs.

The three parts of the SLS-Fusion algorithm are depth estimation, conversion of data representation, and LiDAR-based 3D object detection. The depth estimation part is shown in Figure 1. It uses a pair of stereo images and the re-projected depth map of any type of LiDAR on both images as inputs.

The model combines stereo images (I_l, I_r) with the corresponding stereo images (S_l, S_r) generated by re-projecting LiDAR. Both images and point clouds are extracted using an encoder-decoder network. The backbone is inspired by that of DeepLiDAR [30]. Instead of only using left and right images as in [4] and [3], the proposed network uses a weight-sharing pipeline for both LiDAR and images (I_l and S_l) and (I_r and S_r). Following the decoding, the left and right features are fed into the Depth Cost Volume (DeCV) found in [3] to calculate the depth information loss. Like in [3], we use the smooth $L1_{loss}$ function

$$\sum_{(u,v)\in I_l} |d(u,v) - D(u,v)|\,, \tag{1}$$

where a valid depth ground truth is denoted by $d(u,v)$. The depth map is denoted by D, where $D(u,v)$ represents the depth corresponding to pixel (u,v) on the left image I_l. Based on a pinhole model, pseudo point clouds are generated. Given the depth $D(u,v)$ and camera intrinsic matrix, the 3D position

(X_c, Y_c, Z_c) in the camera coordinate system for each pixel (u, v) is given by

$$\text{(depth) } Z_c = D(u, v), \tag{2a}$$

$$\text{(width) } X_c = \frac{(u - c_U) \times Z_c}{f_U}, \tag{2b}$$

$$\text{(height) } Y_c = \frac{(v - c_V) \times Z_c}{f_V}, \tag{2c}$$

where c_U and c_V are the coordinates of the principal point and f_U and f_V are the focal length in pixel width and height, respectively. Following [3], LiDAR is used to improve the quality of the pseudo point cloud. Then, each point $(X_c, Y_c, Z_c, 1)$ is transformed into $(X_l, Y_l, Z_l, 1)$ in the LiDAR coordinate system (the real world coordinate system). Given the camera extrinsic matrix $C = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$, where R and t are respectively the rotation matrix and the translation vector. The pseudo point cloud can be determined as follows

$$\begin{bmatrix} X_l \\ Y_l \\ Z_l \\ 1 \end{bmatrix} = C^{-1} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}. \tag{3}$$

Once the pseudo point cloud is obtained, it can be treated as a normal point cloud, although its accuracy depends on the quality of the predicted depth. Similar to Pseudo-LiDAR++ [3], the input (point clouds) is used to correct errors in the pseudo point cloud. This is a refinement step to obtain a more accurate point cloud. Then the depth map is converted to a pseudo point cloud. The idea is to leverage the power of the top leading LiDAR-based methods such as PointRCNN [5] to eventually detect objects.

B. SLS-Fusion utilization for LiDAR and camera separately

Like [6] and [3], SLS-Fusion [2] involves 3 main steps: depth estimation, data conversion, and LiDAR-based 3D object detection. In our experiments, we try to analyze the contribution of camera and LiDAR for the 3D object detection task. Thus, the two sensors are considered separately. Figure 2

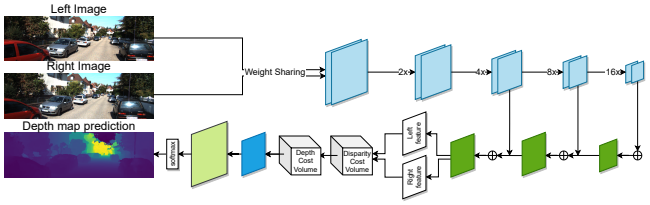

Fig. 2: Modified part of the depth estimation of the SLS-Fusion algorithm to adapt it for using only the camera as input.

shows the network representation for the stereo camera only. We start from the global representation in Figure 1 and consider the depth estimation task. In this global representation, we indeed have 2 inputs to set up the depth map estimation: one for the LiDAR and one for the stereo camera. As it can be seen in Figures 2 and 3, in the networks for the LiDAR

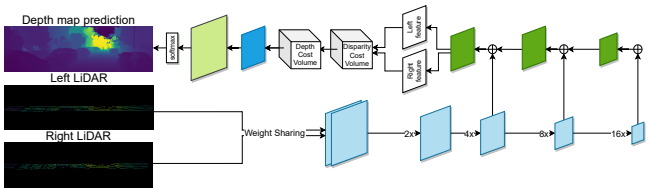

Fig. 3: Modified part of the depth estimation of the SLS-Fusion algorithm to adapt it for using only the LiDAR as input.

and the stereo camera, the part that allows the prediction of the depth maps changes. The last step, obstacle detection, remains the same whether the depth map obtained from LiDAR or the depth map obtained from the stereo camera is used. This is the method used in SLS-Fusion.

As can be seen in Figure 2, the LiDAR has been eliminated. Thus, the output depth map is based on the stereo camera as input. In contrast, Figure 3 shows the model that uses LiDAR only. In this case, the input data is the LiDAR and the output depth map is based on the LiDAR only.

V. Experiments

A. Evaluation Metrics

Common metrics [7], [8] are used to evaluate object detection algorithms. The Average Precision (AP) for both 3D and Bird's Eye View (BEV) is reported as AP_{3D} and AP_{BEV}, respectively, with thresholds of 0.5 and 0.7 for Intersection over Union (IoU). The objects are classified into three difficulty levels: Easy, Moderate and Hard, depending on the size of the 2D bounding box, the occlusion, and the degree of truncation of the object appearing in the RGB image according to [8]. The experiments are performed using the KITTI [8] and Multifog KITTI [1] datasets. The original 3D object detection model SLS-Fusion [2] and its modified version presented in Section IV are used for all experiments.

B. Experimental Protocols

It has been shown that the combination of LiDAR and camera does not outperform the results of models using only LiDAR in normal weather conditions (without rain, fog). On the other hand, LiDAR and camera data are strongly affected by noise in foggy weather. In this case, we do not know what is the best choice: to continue to fuse the two sensors or to use them separately. That is the question we want to answer here. We perform the following experiments: we test the SLS-Fusion model with only camera or LiDAR inputs to see how each sensor contributes to the performance of the model when the data is affected by fog. To analyse the contribution of each sensor, we first compare the results on the KITTI dataset with those on the Multifog KITTI dataset. Knowing that the LiDAR gives poor results in foggy weather, we try to see if a LiDAR with more laser beams (64 instead of 4) would give better results in those adverse conditions.

When analysing fog data, the visibility interval is included in the data. For each level of visibility (level 1 to level 6), object detection performance is considered. This can then

determine up to what visibility range the models can provide acceptable performance.

C. Implementation Details

The procedure used is quite similar to the common pipeline of Pseudo-LiDAR [6].

The depth estimation network is implemented in Pytorch. To achieve faster convergence, complete depth maps from the Scene Flow dataset are first used for training. Fine-tuning is then performed on the 3,712 training samples in the KITTI dataset for 100 epochs, with the batch size set to 4 and the learning rate set to 0.001. Work reported by You *et al.* [3] is followed to generate the simulated 4 beams laser LiDAR (which is as similar as possible to the 4 beams sensor called ScaLa) from the 64 beams LiDAR and then project it onto the left and right image planes to feed into the depth estimation network.

To make the pseudo point cloud closer to the real LiDAR signals, following [6], reflectance is set to 1 and points higher than 3 m from the ground are removed. Then the LiDAR-based 3D object detector can be applied to these remaining points.

For 3D object detection, PointRCNN [5], a LiDAR-based method with a high performance, is used as a basis by many other methods. The method is developed to consider the sparse point cloud. In order to use the pseudo point cloud, we have to make a pre-processing. This latter consists in subsampling the pseudo point cloud in order to obtain a sparser point cloud compatible with a 64 beams LiDAR. Then, the released implementations of PointRCNN are directly used, and their guidelines followed to train it on the training set of KITTI object detection dataset only for the class "Car", since car is one of the main objects and occupies the largest percentage in KITTI dataset, which causes imbalance between "Car" and other classes.

VI. Results and Discussion

As shown in Tables I, II and III, the results of different tests using the SLS-Fusion method are shown for IoU (Intersection over Union) of 0.5 and 0.7, respectively. In each cell of these tables, a pair of numbers A/B corresponds to the results obtained with the AP_{BEV}/AP_{3D} metrics on the KITTI or Multifog KITTI datasets. In describing the results, we have considered the AP_{3D} metric.

In Table I, the experiments are divided into 2 parts. The upper part consists of experiments (1.1, 1.2, 1.3 and 1.4) on the KITTI dataset. In contrast, the experiments in the lower part (2.1, 2.2, 2.3, 2.4 and 2.5) were performed on the Multifog KITTI dataset.

The upper part 1.1 reports the experiment mentioned in [2]. It is a test of the SLS-Fusion model, with a stereo camera and a 4 beams LiDAR fused (L4+S) as inputs experimented on the KITTI dataset. In the next two experiments, we present the results in the same way, but 1.2 only takes a 4 beams LiDAR as input (L4) and 1.3 only takes stereo camera as input (S). In this part which concerns using the KITTI dataset without fog, one notices, as expected, that the fusion of the stereo camera and LiDAR provides good results **93.02%** for Easy and IoU of 0.5. In this configuration, a stereo camera provides better results than a 4 beams LiDAR (**89.39%** versus **78.16%**). Separating the two sensors results in lower performance. This prompts us to consider the two sensors jointly. On the other hand, if we consider the 64 beams LiDAR (line 1.4), the results are very promising (**97.3%** for Easy objects and an IOU of 0.5). Nonetheless, the 64 beams LiDAR price is very high compared to that of cameras and 4 beams LiDARs.

In the second part of Table I, we calculated the same indicators as before, but on data with fog (Multifog KITTI). Performance when fusing a stereo camera and a 4 beams LiDAR (L4+S) remains high (**90.15%**) for Easy objects, which demonstrates the viability of using both sensors at the same time. The performance of the 4 beams LiDAR alone (L4) decreases sharply (**13.43%**), which seems to show that the 4 beams LiDAR is not useful in fog, and it is the stereo camera (S) that has the highest contribution, as its performance remains very high (**89.36%**). The combination of stereo camera and 64 beams LiDAR provides good results (**89.26%** for Easy objects) and for most of the types of objects, but the stereo camera still plays the most important role.

Tables II, III and IV, show results on AP and according to the visibility distances (from level 1 to level 6). In this configuration, we consider 3D detection performance by combining a stereo camera and a 64 beams LiDAR analyzed by visibility distance. For the Easy category (IoU of 0.5), the combination of the two sensors (S+L64) leads to an AP greater than **82.52%**. On the other hand, for the Moderate category (with an IoU of 0.7), visibility distance plays a role, and we have an AP gain of about **7.4%** (difference between **46.71%** at level 1 and **54.10%** at level 6).

If we now look at the sensors separately and depending on the visibility distance, we see the following. For the 64 beams LiDAR (Table III), objects classified as Easy (with an IoU of 0.5) are detected at **71.11%** for visibility level of 20 m (level 1), up to **84.95%** for level 6. On the other hand, if we consider Moderate objects (with an IoU of 0.7) we see that at low visibility level 1, the AP is **17.24%**. Conversely, the AP at level 6 is **35.19%**. We see that the results are better when the visibility is better. However, the two results are bad given the safety requirements of autonomous vehicles. All this shows that in the presence of fog, the use of a LiDAR alone does not provide satisfactory results.

Table IV provides the results for the stereo camera taken into account alone, in foggy weather and as a function of the visibility distance. For Easy objects (IoU of 0.5) we notice that the AP does not vary much depending on visibility (maximum **96.17%** of AP). If we consider Moderate objects (with an IoU of 0.7) we can note that the AP varies according to the visibility distance, ranging from **46.70%** to **54.09%**. If we compare the results to those of a 64 beams LiDAR, we can clearly see the superiority of the stereo camera. Therefore, in foggy weather, LiDAR alone is not suitable.

Idx	Method	Dataset	Input	0.5 IoU			0.7 IoU		
				Easy	Moderate	Hard	Easy	Moderate	Hard
1.1	SLS-Fusion	KITTI	L4+S	93.16/ **93.02**	88.81/ 86.19	83.35/ 84.02	87.51/ 76.67	76.88/ **63.90**	73.55/56.78
1.2	SLS-Fusion	KITTI	L4	84.02/ **78.16**	72.98/ 68.92	66.34/ 63.92	56.72/ 38.82	49.25/ 32.02	44.14/ 29.75
1.3	SLS-Fusion	KITTI	S	89.50/ **89.39**	78.54/ 77.46	75.19/ 69.77	82.21/ 66.54	62.18/ 47.18	56.41/ 43.07
1.4	PointRCNN	KITTI	L64	97.3/ **97.3**	89.9/ 89.8	89.4/ 89.3	90.2/ 89.2	87.9/ 78.9	85.5/ 77.9
2.1	SLS-Fusion	Multifog KITTI	L4+S	90.27/ **90.15**	79.17/ 78.01	76.12/ 70.21	83.42/69.57	62.79/ 48.19	56.84/44.85
2.2	SLS-Fusion	Multifog KITTI	L4	15.44/ **13.43**	10.63/ 9.58	9.75/ 9.65	10.87/ 9.09	9.09/ 9.09	9.09/ 9.09
2.3	SLS-Fusion	Multifog KITTI	S	89.52/ **89.36**	78.75/ 77.71	75.63/ 69.90	82.41/ 70.52	62.59/ **48.27**	57.11/ 45.75
2.4	SLS-Fusion	Multifog KITTI	L64	81.61/ 77.31	56.89/ 53.87	49.83/ 47.80	58.35/ 42.57	38.80/ 29.27	34.63/ 25.25
2.5	SLS-Fusion	Multifog KITTI	L64+S	89.42/ **89.26**	78.79/ 77.82	75.92/ 74.58	82.85/ 71.49	62.33/ 48.39	57.10/ 45.78

TABLE I: AP_{BEV}/ AP_{3D} results on the KITTI dataset for the category "Car" with IoU at 0.5 and 0.7 and on three levels of difficulty: Easy, Moderate, and Hard. S, L4, L64 denote the stereo camera, the 4 beams LiDAR, and the 64 beams LiDAR, respectively.

Visibility	Num obj (train)	Num obj (test)	0.5 IoU			0.7 IoU		
			Easy	Moderate	Hard	Easy	Moderate	Hard
Level 1	2,363	2,200	88.95/ 88.68	75.29/ 69.78	69.59/ 68.15	77.05/ 64.99	57.15/ **46.71**	54.83/ 44.05
Level 2	2,381	2,240	96.42/ 96.16	78.98/ 78.31	75.96/ 74.65	86.36/ 75.49	63.58/ 52.68	57.76/ 46.22
Level 3	2,249	2,369	89.24/ 89.03	78.15/ 77.27	76.45/ 74.37	83.91/ 71.51	62.69/ 47.98	57.19/ 45.49
Level 4	2,343	2,536	84.30/ **82.52**	59.41/ 57.83	54.55/ 50.24	62.08/ 47.80	40.77/ 30.85	35.60/ 27.24
Level 5	2,353	2,372	89.85/ 89.75	83.31/ 77.86	76.68/ 74.60	85.88/ 74.76	62.98/ 52.52	56.85/ 46.52
Level 6	2,668	2,668	89.68/ 89.46	78.88/ 77.94	76.93/ 74.85	85.70/ 74.79	64.95/ **54.10**	58.28/ 48.12

TABLE II: Detailed results for each fog density. The SLS-Fusion algorithm is applied to the Multifog KITTI dataset. It takes stereo camera and 64 beams LiDAR as inputs in this case.

Visibility	Num obj (train)	Num obj (test)	0.5 IoU			0.7 IoU		
			Easy	Moderate	Hard	Easy	Moderate	Hard
Level 1	2,363	2,200	73.64/ 71.11	48.72/ 45.54	44.73/ 40.27	43.84/ 26.12	27.55/ **17.24**	26.30/ 16.94
Level 2	2,381	2,240	85.46/ 77.42	57.86/ 53.62	50.05/ 47.06	57.04/ 41.07	37.90/ 28.04	33.82/ 24.76
Level 3	2,249	2,369	83.38/ 78.72	57.60/ 54.59	50.38/ 48.69	56.03/ 43.49	38.18/ 29.29	32.00/ 25.71
Level 4	2,343	2,536	89.88/ 89.78	84.50/ 79.39	78.54/ 77.13	84.27/ 72.15	64.99/ 54.63	57.48/ 47.27
Level 5	2,353	2,372	83.47/ 82.84	58.26/ 57.17	50.44/ 49.29	66.79/ 54.62	42.25/ 34.84	36.83/ 30.28
Level 6	2,668	2,668	86.24/ **84.95**	59.87/ 58.01	56.14/ 51.38	65.77/ 51.37	41.95/ **35.19**	39.42/ 30.89

TABLE III: Detailed results for each fog density. The SLS-Fusion algorithm is applied to the Multifog KITTI dataset. It takes 64 beams LiDAR as input in this case.

Visibility	Num obj (train)	Num obj (test)	0.5 IoU			0.7 IoU		
			Easy	Moderate	Hard	Easy	Moderate	Hard
Level 1	2,363	2,200	89.02/ 88.77	75.19/ 69.56	69.41/ 68.10	77.02/ 64.85	57.11/ **46.70**	54.82/ 44.03
Level 2	2,381	2,240	96.47/ **96.17**	78.97/ 78.29	75.91/ 74.66	86.31/ 75.45	63.57/ 52.66	57.75/ 46.20
Level 3	2,249	2,369	89.25/ 89.05	78.14/ 77.25	76.44/ 74.33	83.88/ 71.50	62.66/ 47.97	57.20/ 45.50
Level 4	2,343	2,536	84.33/ 82.55	59.40/ 57.81	54.53/ 50.20	62.07/ 47.79	40.75/ 30.84	35.60/ 27.23
Level 5	2,353	2,372	89.86/ 89.77	83.30/ 77.85	76.65/ 74.59	85.86/ 74.75	62.97/ 52.50	56.83/ 46.50
Level 6	2,668	2,668	89.70/ 89.47	78.85/ 77.92	76.92/ 74.84	85.68/ 74.77	64.93/ **54.09**	58.26/ 48.10

TABLE IV: Detailed results for each fog density. The SLS-Fusion algorithm is applied to the Multifog KITTI dataset. It takes stereo camera as input in this case.

VII. Conclusion

In this study, we have tested the capabilities of 3D obstacle detection using two types of sensors: two versions of LiDAR (4 beams and 64 beams) and a stereo camera. The developed algorithms were tested on the KITTI dataset and then with additional fog (Multifog KITTI). Based on obstacle detection performance, we have analyzed several aspects: the contribution of the two types of sensors both in normal weather and in fog, when they are combined and when they are used separately. The main result is that using LiDAR in foggy weather leads to a slightly worse obstacle detection performance (even worse when the LiDAR is a 4 beams laser sensor). On the other hand, results based on stereo camera are promising in foggy weather, regardless of level of visibility.

The results in Tables I, II, III and IV contain a lot of information that needs to be interpreted in detail. In the context of this article, and given the limited number of pages, we presented summarised results, with main tendencies, and we were not able to go in more depth in the interpretation. We plan to do this shortly in a separate publication.

As a reminder, the results of this study are obtained on real LiDAR and camera data, initially acquired in clear weather,

to which fog has been added. The model used is a simple model, which considers only the macroscopic attenuation phenomenon. This model has been calibrated on tests performed in a controlled environment on standard sensors [10]. However, this model has two limitations. (i) First, it should be verified that it is valid for the sensors used in the KITTI dataset, because, depending on the sensor (brand, type, internal settings), the impact of fog can be more or less strong. In the case of the LiDAR, the threshold effects when passing from the raw signals to the point cloud can have a strong impact on the model we use. For the camera, the exposure setting is not considered here, and may once again have an impact not modeled here. (ii) The model used here does not consider the microscopic phenomena of light diffusion. Thus, the halo effects for the camera and the backscattering effects for a LiDAR sensor are not simulated here. These different elements are known limitations of the model and clearly explained. They can have an impact on the results, so the results presented here should not be taken as categorical, but as initial results allowing to compare sensors, and to find data fusion solutions adapted to the autonomous vehicle.

For future work, two options can be considered to circumvent the limitations of the model used here. The first would be to perform acquisitions on real site, under real adverse weather conditions. The second, more promising option, would be to use more complex 3D models that implement microscopic scattering phenomena and simulate the complete path of light from objects to the sensor and the sensor itself.

REFERENCES

[1] N.A.M. Mai, P. Duthon, L. Khoudour, A. Crouzil, S.A. Velastin, "3D Object Detection with SLS-Fusion Network in Foggy Weather Conditions," Sensors, 2021.

[2] N.A.M. Mai, P. Duthon, L. Khoudour, A. Crouzil, S.A. Velastin, "Sparse LiDAR and Stereo Fusion (SLS-Fusion) for Depth Estimation and 3D Object Detection," in ICPRS, pp. 150-156(6), 2021.

[3] Y. You, Y. Wang, W.L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, K.Q Weinberger, "Pseudo-LiDAR++: Accurate depth for 3D object detection in autonomous driving," in ICLR, 2020.

[4] J.R. Chang, Y.S. Chen, "Pyramid Stereo Matching Network," in CVPR, pp. 5410–5418, 2018.

[5] S. Shi, X. Wang, H. Li, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," in CVPR, pp. 770–779, 2019.

[6] Y. Wang, W.L. Chao, D. Garg, B. Hariharan, M. Campbell, K.Q Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in ICCV Workshops, 2019.

[7] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, "The Pascal visual object classes (VOC) challenge," in Springer, pp. 303-338, 2010.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," in IJRR, vol. 32, no. 11, pp. 1231–1237, 2013.

[9] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in CVPR, pp. 4040–4048, 2016.

[10] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, F. Heide, "Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather," in CVPR, pp. 11679-11689, 2020.

[11] B. Yang, W. Luo, R. Urtasun, "Pixor: Real-time 3D object detection from point clouds," in CVPR, pp. 7652–7660, 2018.

[12] Y. Yan, Y. Mao, B. Li, "SECOND: Sparsely Embedded Convolutional Detection," Sensors, 2018.

[13] A. H Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in CVPR, pp. 12689-12697, 2019.

[14] Z. Yang, Y. Sun, S. Liu, J. Jia, "3DSSD: Point-based 3D single stage object detector," in CVPR, pp. 11040–11048, 2020.

[15] C. He, H. Zeng, J. Huang, X.-S. Hua, L. Zhang, "Structure aware single-stage 3D object detection from point cloud," in CVPR, pp. 11873–11882, 2020.

[16] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in NIPS, pp. 91–99, 2015.

[17] K. He, G. Gkioxari, P. Doll'ar, R. Girshick, "Mask R-CNN," in ICCV, pp. 2961–2969, 2017.

[18] X. Chen, K. Kundu, Y. Zhu, A. G Berneshawi, H. Ma, S. Fidler, R. Urtasun, "3D object proposals for accurate object class detection," in NIPS, pp. 424-432, 2015.

[19] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, T. Chateau, "Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image," in CVPR, pp. 2040–2049, 2017.

[20] A. Mousavian, D. Anguelov, J. Flynn, J. Kosecka, "3D bounding box estimation using deep learning and geometry," in CVPR, pp. 7074-7082, 2017.

[21] B. Xu, Z. Chen, "Multi-level fusion based 3d object detection from monocular images," in CVPR, pp. 2345–2353, 2018.

[22] P. Li, X. Chen, S. Shen, "Stereo R-CNN based 3D object detection for autonomous driving," in CVPR, pp. 7644–7652, 2019.

[23] Z. Qin, J. Wang, Y. Lu, "Triangulation learning network: from monocular to stereo 3D object detection," in CVPR, pp. 7615–7623, 2019.

[24] C. Li, J. Ku, S. L. Waslander, "Confidence guided stereo 3D object detection with split depth estimation," in IROS, pp. 5776–5783, 2020.

[25] A. D. Pon, J. Ku, C. Li, S. L. Waslander, "Object-centric stereo matching for 3D object detection," in ICRA, pp. 8383–8389, 2020.

[26] C. R. Qi, W. Liu, C. Wu, H. Su, L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in CVPR, pp. 918–927, 2018.

[27] Z. Wang, K. Jia, "Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection," in IROS, pp. 1742–1749, 2019.

[28] S. Vora, A. H Lang, B. Helou, O. Beijbom, "Pointpainting: Sequential fusion for 3D object detection," in CVPR, pp. 4604–4612, 2020.

[29] X. Chen, H. Ma, J. Wan, B. Li, T. Xia, "Multi-view 3d object detection network for autonomous driving," in CVPR, pp. 1907–1915, 2017.

[30] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, M. Pollefeys, "DeepLiDAR: Deep surface normal guided depth prediction for outdoor scene from sparse LiDAR data and single color image," in CVPR, pp. 3313–3322, 2019.

[31] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, W. Stork, "Weather Influence and Classification with Automotive Lidar Sensors," in IEEE Intelligent Vehicles Symposium (IV), pp. 1527–1534, 2019.

[32] M. Kutila, P. Pyykönen, H. Holzhüter, M. Colomb, P. Duthon, "Automotive LiDAR performance verification in fog and rain," in ITSC, pp. 1695–1701, 2018.