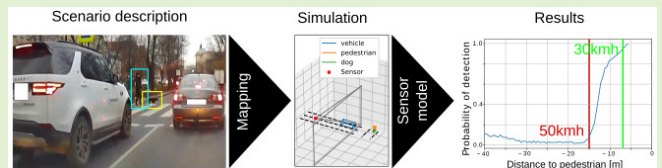


Generic Sensor Model Usecase Exemplified by Pedestrian Crossing

Kamil Lelowicz¹, Member, IEEE, and Adam Krzysztof Piłat¹, Member, IEEE

Abstract—This article presents the application of a generic sensor model (GSM) used in the virtual validation of an advanced driver-assistance system. The aim is to demonstrate the feasibility and usefulness of such a model in a case study. This model transforms an incoming object list from a virtual simulation to a sensor-specific object list and applies to all perception sensors acting at the object level. The procedure for parameterization of the sensor model was explained and illustrated with examples of two algorithms: vision- and lidar-based. The validity of the model was confirmed using the case of a pedestrian crossing where incidents involving pedestrians occur. Based on simulation studies and the evaluation of the actual configuration of the pedestrian crossing, conclusions were drawn which may contribute to the increase of pedestrians' safety, both on the passive and active sides. The obtained results allow for the improvement of the operation of the transportation system in the field of advanced driver-assistance system (ADAS) development.

Index Terms—Mathematical modeling, sensor model, virtual validation.



I. INTRODUCTION

A KEY component in the advanced driver-assistance system (ADAS) is 3-D object detection algorithms. They are currently under heavy development. Novel algorithms employ deep neural networks to solve 3-D detection problems [1], [2], [3], [4]. The use of ADAS functions instead of fully human-controlled vehicles should reduce the number of fatal accidents. About 90% of accidents are caused fully or in part by human error [5]. Assuming that there are 40 000 fatalities on the roads in the European Union, 36 000 deaths are caused by human error [6], [7].

Thus, one of the key challenges is testing ADAS functionality. Algorithms are immensely complex because they process a large amount of data from sensors like cameras, radar, and lidar. To train an algorithm and to prove its robustness in various difficult conditions, one should gather and label data from hundreds of millions of miles of test drives [8], [9]. It is very costly and time-consuming to carry out so many test drives to cover such a great number of kilometers in various conditions. Contrary to real test drives, virtual testing allows for completing millions of kilometers under repeatable conditions. Therefore, virtual

validation is a critical topic in the process of development of ADAS algorithms.

II. RELATED WORK

The general functionality of ADAS can be described by the principle “sense—plan—act,” which originates from robotics [10]. First, information is received (sense). Based on this information, a behavior is planned (plan), which is then transformed into an action (act). In order to be able to operate the ADAS functionality in the same way as it would function in the actual vehicle, it is integrated into the simulation system. The system provides all necessary interfaces to other vehicle systems as well as interfaces to sensor technology for the perception of the vehicle environment. The interaction between the virtual world generated by the simulation system and the ADAS function is modeled by sensor models. Consequently, sensor models are a prerequisite for virtual testing. Virtual environment [11] like CARLA [12], IPG Carmaker [13], and ASM traffic [14] simulate the scenario and includes the necessary perception information for environmental sensors. The ground truth, the true state of the environment, propagated to the sensor model. The role of the sensor model is to capture the relationship between sensor output and ground truth as realistically as possible. The sensor model processes the perceptual information and forwards it to ADAS algorithms, which are responsible for vehicle control: steering, acceleration, and braking. The future and critical application of the ADAS system with well-realized sensing of the surrounding is the mono-cycle [15], [16]. The vehicle dynamics model converts control values into vehicle movement. The components and their interaction are shown in Fig. 1

Manuscript received 31 August 2022; accepted 26 September 2022. Date of publication 6 October 2022; date of current version 14 November 2022. This work was supported by the Polish Ministry of Science and Higher Education under Project 0014/DW/2018/02. The associate editor coordinating the review of this article and approving it for publication was Dr. Guofa Li. (Corresponding author: Adam Krzysztof Piłat.)

The authors are with the Department of Automatic Control and Robotics, AGH University of Science and Technology, 30-059 Kraków, Poland (e-mail: lelowicz@agh.edu.pl; ap@agh.edu.pl).

Digital Object Identifier 10.1109/JSEN.2022.3211092

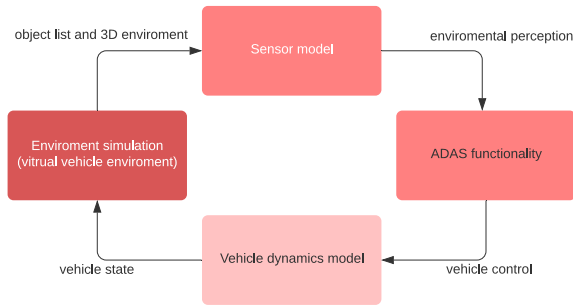


Fig. 1. Simulation setup for development of automated driving systems. A sensor model provides the environmental perception.

Typically sensor models are classified into three categories: ideal, probabilistic, and physical, or low-fidelity, medium-fidelity, and high-fidelity [6], [17]. Ideal sensors simulate zero-error and perfect sensors, which correctly recognize all objects in a field of view (FOV) [18]. Probabilistic sensor models set up a probabilistic function $p(\text{output}|\text{ground-truth})$ between the ground truth and sensor output [19], [20]. The influence of the environment and sensor error is modeled based on statistics and observation. The output from both the ideal model and the probabilistic model is a list containing a detected object. On the other hand, output from the physical sensor is raw measurement generated by physical laws [21], [22]. To achieve that, usually rendering and ray-tracing methods are used. Moreover, there are models that are a mixture of types, that is, physical and stochastic aspects are included [23], [24].

The object list models are usually called generic sensor models (GSMs) or configurable sensor models, because they can be applicable to sensors like cameras, lidar, and radar. The input to the model is the object list that contains object properties like object class, position, orientation, size, velocity, braking lights status, and so on. Data are processed by a chain consisting of n modules. The properties of the sensor are mapped on sensor effects. Each module describes a modifiable property of a sensor, that is, the object-measured properties can be adjusted according to the algorithm used in the applied model. The output is a filtered list of objects that contains only objects detected by the sensor. Fig. 2 presents a functional overview of a GSM. It also shows which elements of the real system are imitated by the GSM. In this case, the behavior of the whole perception system, including the sensor and perception algorithms, is modeled.

The properties of a sensor's effect can be distinguished into several categories. There are three major categories [25].

- 1) Properties of sensor physical measurement process such as FOV, occlusion, or absorption.
- 2) Signal processing and hardware-based properties such as resolution, latencies, or noise.
- 3) Information processing and interpretation based on properties, for example, object classification, position estimation, or tracking.

In literature GSMs are described in [17], [18], [25], [26], [27], and [28]. The work [18] presents an efficient way to handle the complex sensor's FOV. Borders can be defined by a user. However, only 0-D objects are considered, and

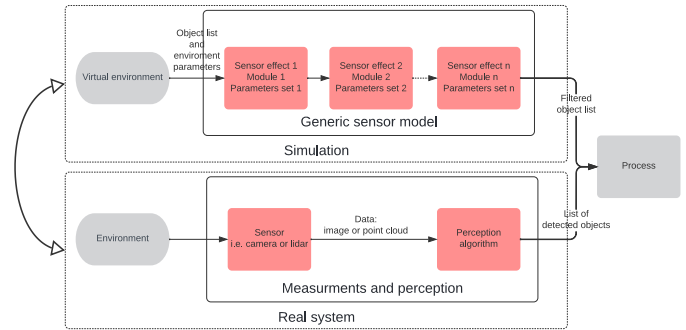


Fig. 2. Functional perspective of the GSM architecture.

occlusion is modeled in a straightforward way by reducing the view angle. An object is detected if the ratio of the uncovered angle and the reduced angle is still above a defined threshold.

The focus of [25] and [26] is to define a modular architecture that describes a way how to filter the object list according to different sensor effects. Additionally, work [25] presents the logical perspective and implementation technical perspective of a generic/configurable sensor model. The usability of the sensor model is demonstrated by testing adaptive cruise control (ACC) in a virtual simulation.

The work [17] considers a model with object class-specific FOVs. Object occlusion is included in the calculation but the assumption is that the world is only 2-D. The detection probabilities assigned to each object class can be used. However, there is no explanation of how the probabilities are calculated and assigned to each object.

The article [27] presents a phenomenological lane detection model to simulate camera performance. The model is in the form of a feedforward neural network. The inputs include vehicle dynamics signals such as speed, acceleration, and the distance between the real trajectory of the vehicle and the centerline of the road. The model estimates the lateral distance from the centerline of the host vehicle to the left/right lane marking and the vehicle heading relative to the lane heading. The results demonstrate that the phenomenological model can sufficiently replicate camera detection performance in the simulation.

The last article [28] presents our solution. The ray-casting-based algorithm is employed to determine object projection size and visibility ratio. The calculation is made in 3-D where each object is described by a 3-D bounding box (3-DBB). The number of casted rays can be easily adopted to match the resolution of the camera or lidar. This makes the solution easily customizable to the physical characteristics of the camera and lidar. Additionally, a single object can be described by several 3-DBB and therefore can better map the shapes of the object. Moreover, the model can be used to assess the quality of object detection algorithms. One can easily filter out several corner cases. For example, a large object that is 100% visible and was not detected by the algorithm or an object that is 100% occluded and is detected. This can also be used, for example, to find errors in the labeling of the scene. Furthermore, based on the statistics of the calculated object

visibility values and its projection size, a probabilistic model of the object detection algorithm can be established.

III. MOTIVATION

Urban mobility research and government initiatives mainly focus on motor vehicle issues, yet pedestrians are one of the most vulnerable elements of the urban landscape. The topic of their safety was often overlooked in the past [29]. In 2020, around 18 800 persons were killed in road accidents in the EU: 44.2% of the fatalities were passenger car drivers or passengers; 19.2% were pedestrians. Additionally, there were 5977 pedestrians killed in traffic crashes in the United States in 2017. Moreover, approximately 137 000 pedestrians were treated in emergency departments for nonfatal crash-related injuries in the United States in 2017. Currently, research on pedestrians focuses on the prediction of their trajectory [30], [31] or a system that is able to create a fine-grained map of hazard levels across a city and a heuristic to identify interventions that might simultaneously improve pedestrian and vehicle safety [29]. This study focuses on the topic of pedestrian crossing safety and was inspired by a real road incident witnessed by one of the authors. A pedestrian walking across a pedestrian crossing was hit by a motorcyclist traveling in one lane. The pedestrian crossing intersects three lanes, including the rightmost lane reserved for buses, electric cars, and emergency vehicles. The incident took place in a traffic situation of significant intensity, as a result of which vehicles stopped in both lanes in order to allow pedestrians to cross the road. The perpetrator of the incident was a motorcyclist who was moving on the bus lane. He spotted the pedestrian at the last moment but started to brake too late, due to which the pedestrian was hit. The immediate cause of the incident was the limited visibility caused by vehicles standing in front of the pedestrian crossing. Thus, the following questions are posed: what information should a vehicle assist with automation systems have and what controls informing the driver or taking actions in the vehicle should develop a decision-making system to make the right decisions in such critical situations in the future? The research considered the following zebra crossing as an illustrative example [see Fig. 3]. The distances L1, L2, and L3 are equal to 2, 58, and 40 m, respectively. The road signs S1, S2, and S3 are located on the left and right sides of the road, as well as horizontally—painted on the road. Additionally, the give way line is not required in many countries. In Poland, it is only recommended for places where there is no intersection with other roads, and the law requires it to be at least 2 m from a pedestrian crossing (L1 distance) [32]. The upper limit is not defined. In the U.K., the give way line should be located at distances from 1.1 to 3 m to the pedestrian crossing. In special cases, the distance may be extended to 10 m [33].

One can find that the hypothetical stopping distance for a car traveling at 50 kmph is approximately 14 m. Typically, the reaction time calculated as the time elapsed between detection and decision is about 0.8 s. Therefore, the extra 14 m needs to be added to the braking distance. It seems that there is enough time and distance to stop the car if one detects the pedestrian passing the S2 road sign. Unfortunately, there are many other

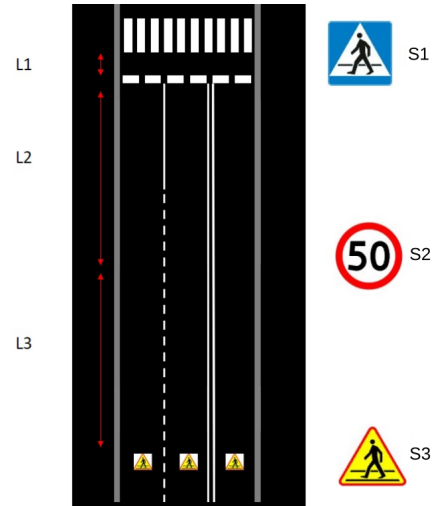


Fig. 3. Zebra crossing configuration.

cars and events interrupting the driver's concentration on the road.

IV. PROBLEM FORMULATION

This article presents the use of the high-level GSM [28] that operates on the object list level. Moreover, the novel approach to the characterization of ADAS 3-D object detection algorithms is presented. The new technique uses high-level information from GSM allowing a description of the probability of object detection by respective algorithms. In addition, it allows the definition of object detection inaccuracies depending on the class and location of the object in relation to the host. Two sample algorithms were investigated: FCOS3D [3] and Centerpoint [2]. The use case is exemplified by a pedestrian crossing. The behavior of the algorithm described by the GSM is tested in a virtual environment. Obtained results are used to discuss the design of pedestrian crossings.

The presented solutions also allow for a more accurate representation of ADAS functionality in high-level simulations to determine, among other things, the effective range of operation of particular functionalities at an early stage of the design of the entire ADAS system. On the basis of the conducted research, new insights and conclusions related to ADAS and pedestrian crossings are given.

V. ALGORITHMS

A. Fully Convolutional One-Stage Monocular 3-D Object Detection

The 2-D information is not enough to correctly perceive the 3-D real world for an intelligent vehicle. Accurate 3-D information about objects and the road around is necessary for a highly automated car to operate safely in traffic. Therefore, 3-D object detection algorithms are becoming increasingly important. Most state-of-the-art algorithms depend on lidar inputs which provide accurate 3-D information about object position. However, the price of lidar solutions and the presence of moving parts make their use in car production limited. Therefore, a monocular camera that can provide 3-D object detection, as a cheaper and simpler solution, becomes a much more meaningful research problem nowadays. In this article,

the focus is on fully convolutional one-stage monocular 3-D object detection (FCOS3D). The solution wins the first place out of all the vision-only methods in the nuScenes 3-D detection challenge of NeurIPS 2020 [34]. The algorithm and code are publicly available at `mm detection3d` [35].

FCOS3D is an example of a fully convolutional one-stage detector consisting of three components: a backbone for feature extraction, a neck for multilevel scaling, and a head for dense output. As a backbone, a pretrained ResNet101 was used. Neck, the next module is based on the feature pyramid network. The last stage, the detection head, follows the conventional design of RetinaNet and FCOS resulting in an anchor-free stage detector.

The algorithm was evaluated on the Nuscense dataset [36] (see <https://www.nuscenes.org>). The validation dataset incorporated 6019 scenes with 117672 labeled objects. The obtained key performance indicators (KPIs) are presented in Table I. The L_2 -norm was used as an association method. Objects were marked as true positive (TP) if the distance between the center of the ground-truth bounding box to the predicted bounding box center was less than 2 m and the object class matched.

KPIs broken down by the detected object class are presented in Table II. It is worth pointing out that the number of samples in class correlates with the results. The proportions of the number of given class objects were more or less the same in the training and validation sets. This clearly shows that there were not enough particular types of objects in the dataset. The best results were obtained for the “car” and “pedestrian” object classes. By contrast, the worst results were achieved for the “heavy” class. The amount of data not only affects the performance of the algorithm, but also the ability to analyze obtained results. When there are few objects for classes, then it is difficult to infer the quality of the algorithm for these classes. The only conclusion that can be drawn is that it is not fully representative, which leads to the consequence that the dataset needs to be supplemented. In our case, only the classes “car,” “pedestrian,” and “general” contain enough objects to be analyzed.

B. Centerpoint

Center-based 3-D object detection and tracking (Centerpoint) is a point cloud-fed algorithm that used output from lidar only as an input [2]. 3DBB is usually used to represent a 3-D object. This representation imitates a well-known 2-D bounding-box detection on an image plane. However, additional dimension generates new challenges. One of the biggest difficulties is to correctly predict the orientation of the object or fit an axis-aligned bounding box to a rotated object. To overcome this problem, Centerpoint represents and detects objects as points.

In the first stage, Centerpoint detects the center of the objects and their properties [37]. To achieve this, a standard lidar-based backbone network like PointPillars [38] or Voxelnet [39] is utilized. Next, the output is flattened into an overhead map view and then standard image-based keypoint detectors are used to find the center of the objects and regress to other attributes. The light-weighted second stage refined

TABLE I
GENERAL KPI FOR FCOS3D

True positive	72553
False negative	45119
False positive	31545
Recall	0.61657
Precision	0.69758
F1	0.65458
mAP	0.50653
Number of scenes	6019.0
Threshold	0.15

these estimates. It extracts point features at the 3-D centers of each face of estimated objects 3DBB. The result of this is a performance increase with minor costs. The general KPIs calculated for Centerpoint are presented in Table III.

Lidar-based algorithms perform much better than the camera-only counterparts [36]. In the case of the algorithms in question, 28% more objects were correctly detected by Centerpoint in comparison to FCOS3D. Moreover, the number of false positives also dropped significantly by 32%. This is related to the threshold that is twice as large in comparison to FCOS3D. The FCOS3D algorithm does not have direct information about the depth of the image. The result is that it generates more or less confident detections that are not correctly located in 3-D space. The results are worse, but it should be remembered that this sensor is several times cheaper and does not contain any moving parts. Thus, its resistance to damage and robustness should be higher, which is also important when validating algorithms.

Detailed results for each class are presented in Table IV. As in the previous case, there is also a correlation between the number of objects of a given class and the obtained results for this class. The best results are again achieved for the classes “car” and “pedestrian.” Nevertheless, the number and variety of individual classes do not allow the algorithm to correctly learn to recognize objects that belong to these classes. The results for “bicycle” are surprising, the algorithm is characterized by excessive eagerness to find these objects in the surrounding environment. In addition, it can be observed that the results for the “general” class are weak despite a large number of objects. This is a class that is very diverse and therefore many examples in the training set are needed to obtain good results. The use of lidar as an input to the algorithm allows for better results with a limited number of objects in the dataset. The best algorithms that use data and information from all sensors such as cameras, lidar, and radar achieve only slightly better results compared to solutions based only on lidar input [36]. This shows that lidar is the best source for algorithms that detect objects in 3-D.

VI. ALGORITHMS ANALYSIS

The probability of detection for each algorithm was derived on the basis of the data coming out of the GSM.

- 1) Visibility ratio.
- 2) Projection size.

These values were determined for all objects included in the analyzed dataset. Next, using the results that were performed during the calculation of the KPIs, each object was marked as

TABLE II
KPI RESULTS PER CLASS FOR FCOS3D

Class	AP	TP	FN	FP	Recall	Precision	F1
car	0.7237	33839	12241	13070	0.73435	0.72138	0.72781
truck	0.47099	4600	4194	4029	0.52308	0.53309	0.52804
pedestrian	0.65249	13692	8923	5743	0.60544	0.7045	0.65122
general	0.63811	18356	15199	6766	0.54704	0.73067	0.62566
bicycle	0.65458	775	1149	538	0.40281	0.59025	0.47884
motor	0.4314	745	1072	348	0.41002	0.68161	0.51203
heavy	0.16084	546	2341	960	0.18912	0.36255	0.24858

TABLE III
GENERAL KPI FOR CENTERPOINT

True positive	93052
False negative	27141
False positive	21549
Recall	0.77419
Precision	0.81196
F1	0.79263
mAP	0.67542
Number of scenes	6019.0
Threshold	0.32

TP, false negative (FN), or false positive (FP). The sensitivity of the algorithm was then calculated for each visibility interval and projection size. In the case of visibility, the interval had a length of 0.1, while for the projection size, the interval was set to 2500. The “car” and “pedestrian” classes were analyzed because of the large number of objects of these classes in the set and because these two classes were used in the simulation of pedestrian crossings. The results obtained are shown as red dots in Figs. 4 and 5 for the FCOS3D algorithm and in Figs. 6 and 7 for the Centerpoint algorithm. Furthermore, using optimization methods, a plane describing the probability of detection was determined on the basis of the previously determined points.

The function describing the detection probability plane was chosen in the following form:

$$f : \langle 0, 1 \rangle \times \mathbb{R}^+ \cup \{0\} \times \mathbb{R}^7 \rightarrow \langle 0, 1 \rangle$$

$$f(x, y, \mathbf{k}) = \max \left\{ \min \left\{ k_1 x^{k_2} y^{k_3} \left(k_4 x^{k_5} + k_6 y^{k_7} \right), 1 \right\}, 0 \right\} \quad (1)$$

where

- x visibility;
- y projection area;
- \mathbf{k} vector of coefficients.

This construction of the function describing the detection probability plane ensures that when the visibility is equal to zero or the projection size is equal to zero, the value of the function is also equal to zero

$$f(0, y, \mathbf{k}) = f(x, 0, \mathbf{k}) = 0 \quad \forall x, y. \quad (2)$$

In addition, the codomain of the function is in the range $\langle 0, 1 \rangle$.

The optimization problem was defined as follows:

$$\min_{\mathbf{k}} c^2 \sum_{j=0}^n \ln \left(1 + \frac{(f(x_j, y_j, \mathbf{k}) - d_j)^2}{c^2} \right) \quad (3)$$

where

- n number of data obtained for the class of object;
- d_j calculated recall within the specified range;
- c empirical coefficient used during optimization process.

The construction of the objective function was designed to severely weaken the influence of outliers. The results of the optimization are shown in Figs. 4–7 using the blue plane.

Compared to the basic KPIs, the presentation of the data in the form of these graphs gives a more complete picture of the behavior of the algorithm. It can be seen that objects that are almost fully visible and their projection size is large enough are detected correctly with almost 100% efficiency. It can also be seen that the Centerpoint algorithm obtains better results. Statistically, it needs less information about an object to make a correct prediction. This can be seen especially in the case of pedestrian detection, where for relatively small objects, the algorithm makes correct predictions. When analyzing these results, it is also worth noting that through occlusions or the fact that an object is at a certain distance, it can be more difficult to detect. These parameters are not considered when calculating basic KPIs such as mAP, recall, or precision. This leads to the conclusion that the importance of a given object and the weighting of its impact on the environment when calculating such KPIs should be taken into account because occluded ones that are in the distance have far less impact on a moving host.

In the case of the estimation error of the position and rotation of the predicted object, it can be noted that they depend mainly on the size of the object projection on the sensor. This is noticeable for both algorithms and both “car” and “pedestrian” classes. The results are shown in Figs. 8 and 9. For car detection, the detection error in the direction perpendicular to the car movement [Figs. 8(b) and 9(b)] is smaller than in the parallel direction [Figs. 8(a) and 9(b)]. The algorithms have a worse estimation capability of the depth at which objects are located than their lateral location. For the FCOS3D algorithm, the mean estimation error of the position x varies from 0.20 to 0.75 m. In contrast, the estimation of the position y has an error ranging from 0.15 to 0.60 m. In the case of the Centerpoint algorithm, the ranges obtained are, respectively, from 0.10 to 0.28 m and from 0.05 to 0.23 m. In all cases,

TABLE IV
KPI RESULTS PER CLASS FOR CENTERPOINT

Class	AP	TP	FN	FP	Recall	Precision	F1
car	0.88484	39256	6824	3625	0.85191	0.91546	0.88254
truck	0.75325	6699	2095	1726	0.76177	0.79513	0.77809
pedestrian	0.85746	20455	3575	3681	0.85123	0.84749	0.84935
general	0.66924	23023	11592	8037	0.66512	0.74124	0.70112
bicycle	0.49628	1106	822	2553	0.57365	0.30227	0.39592
motor	0.66445	1307	553	1416	0.70269	0.47999	0.57037
heavy	0.40239	1206	1680	511	0.41788	0.70239	0.52401

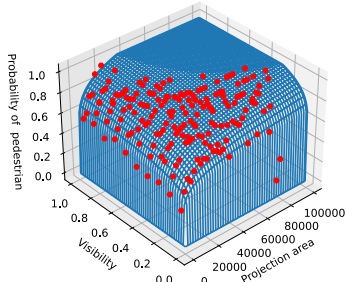


Fig. 4. Probability of pedestrian detection with respect to visibility and projection area for FCOS3D.

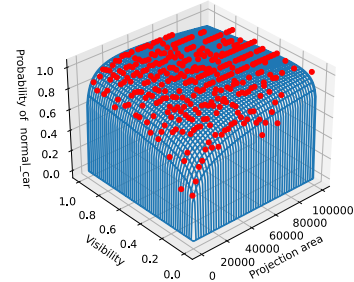


Fig. 7. Probability of normal car detection with respect to visibility and projection area for Centerpoint.

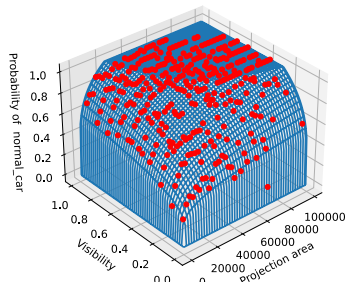


Fig. 5. Probability of normal car detection with respect to visibility and projection area for FCOS3D.

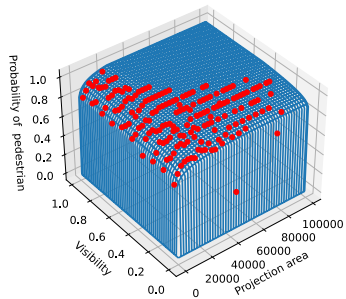


Fig. 6. Probability of pedestrian detection with respect to visibility and projection area for Centerpoint.

the estimation error exponentially decreases with increasing object size and stabilizes at a certain level. In the case of pedestrian detection, there is no noticeable difference between x and y position estimation [Figs. 8(d) and 9(d) and (e)]. This is probably due to the fact that pedestrians are much smaller than cars. In addition, there are no such significant differences in absolute length between width and length. This makes it easier to correctly predict the location of an object. Moreover, the position estimation error decreases with increasing object size only for FCOS3D [Fig. 8(d) and (e)]. The Centerpoint algorithm, on the other hand, shows a constant error that does not begin to change as the object's projection size changes

[Fig. 9(d) and (e)]. As for cars, the algorithm based on lidar data has better accuracy in predicting the position of objects. For Centerpoint, the position prediction error is approximately 0.12 m for x and y . The vision algorithm has an error range from 0.15 m for large objects to 0.60 m for small objects.

Almost all objects detected by the perception algorithms are located on the plane defined by the carriageways. Consequently, one parameter, yaw angle, was used to describe the orientation. The results for yaw estimation are shown in Figs. 8(c) and (f) and 9(c) and (f). Algorithms have considerable difficulty in determining the correct orientation of an object and this is particularly noticeable for pedestrians. The algorithms in this case find the orientation of the pedestrian correctly when its projection size is very large. For cars, the Centerpoint algorithm has an average error of 17° . For FCOS3D, the error was about 30° . The quality of the labeled objects in the dataset can have a large impact on the amount of error obtained. It is important to note that the manual data labeling process will always introduce bias into the actual data. Manually marking the orientation of objects during the ground-truth data generation process is more difficult than marking the location of these objects. This means that the proportion of errors due to uncertainties in labeling will be higher for the yaw angle. However, labeling errors also affect the estimation of object location.

On the whole, the algorithm analysis shows that the algorithm based on lidar data performs much better in a precise location task. Moreover, the GSM can be used successfully to describe and imitate perception algorithms at a high level of abstraction.

VII. PEDESTRIAN CROSSING CASE

From a number of films recorded with a car camera, a situation was selected which is the best illustration for this work. It should be emphasized that the scenario was not

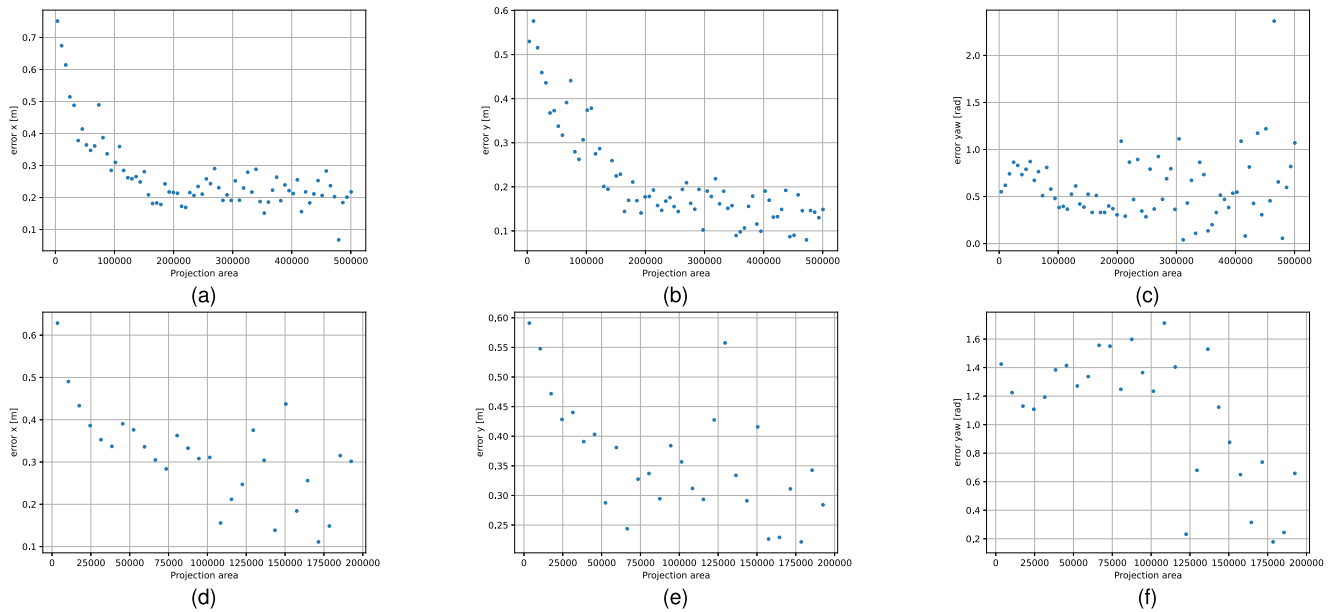


Fig. 8. Location and size errors for detected cars and pedestrians by FCOS3D. (a) Error in estimation car x location with respect to projection size. (b) Error in estimation car y location with respect to projection size. (c) Error in estimation car yaw angle with respect to projection size. (d) Error in estimation pedestrian x location with respect to projection size. (e) Error in estimation pedestrian y location with respect to projection size. (f) Error in estimation pedestrian yaw angle with respect to projection size.

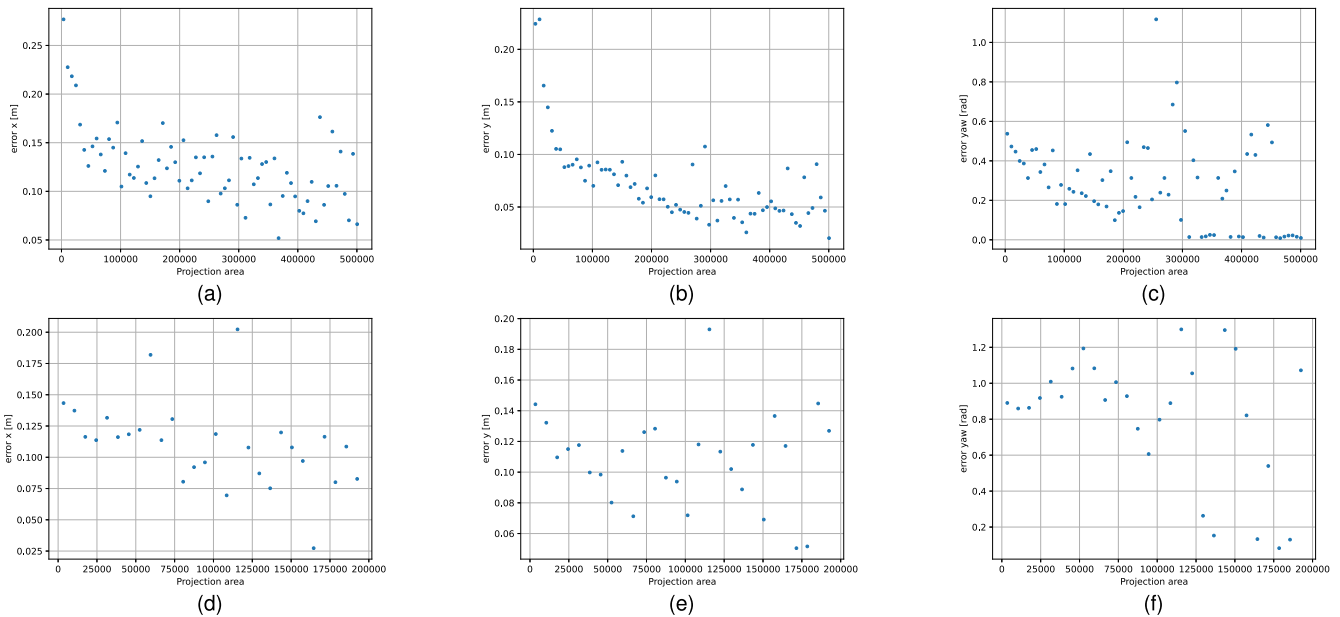


Fig. 9. Location and size errors for detected cars and pedestrians by Centerpoint. (a) Error in estimation car x location with respect to projection size. (b) Error in estimation car y location with respect to projection size. (c) Error in estimation car yaw angle with respect to projection size. (d) Error in estimation pedestrian x location with respect to projection size. (e) Error in estimation pedestrian y location with respect to projection size. (f) Error in estimation pedestrian yaw angle with respect to projection size.

planned. Registration of the scene was completely accidental. For reasons of data protection, the license plates were hidden.

A diagram of the road and its markings is shown in Fig. 3.

It is significant in this situation that the first scene [Fig. 10(a)], shows complete occlusion of objects (in this case, a person and a dog) that are about to cross the pedestrian crossing. It can be seen that the white SUV in the left lane is intended to give way to the crosswalk and took a braking action. The vehicle in the middle lane (preceding the

vehicle recording the event) did not take the braking action early enough, as the objects were occluded and as a result, it entered the pedestrian crossing which is particularly visible in Fig. 10(d). It should be noted that from the point of view of the vehicle recording the event, visibility was also reduced by the occlusion of objects by the SUV. Another important observation is that the animal entered the pedestrian crossing first preceding the entrance of the human [see Fig. 10(b)]. Here, it can also be seen that the person is still occluded



Fig. 10. Zebra crossing—selected scenes to illustrate the occluding and visibility of objects. (a) Scene 1, time 0 s. (b) Scene 2, time 0.3 s. (c) Scene 3, time 0.5 s. (d) Scene 4, time 1.03 s.

by the vehicle in the left lane. Another scene in which the unveiling of the person occurs is illustrated in Fig. 10(c). The time difference between successive scenes is worth noticing. The dynamics of the events is considerable and the times are relatively short (of the order of hundreds of milliseconds). Therefore, the reaction time of the driver must be really short. More than that, it is conservatism, caution, and anticipation that create demands on the characteristics of the driver and the assistance system. These short times increase the demands placed on ADAS systems, which are supposed to work effectively in modern vehicles.

The depicted scene was reproduced in a virtual simulation. The snapshot from the simulation is presented in Fig. 11. The gray lines indicate the FOV of the sensor which is indicated by a red dot. The horizontal FOV was set at 120° , while the vertical was 60° . A car that is occluding the pedestrian and dog is marked with a blue 3DBB. The pedestrian and dog are marked in orange and green, respectively.

The simulation was designed in such a way that the stopping point of a vehicle marked with a blue 3DBB (blue vehicle) was defined (L1 distance). Then 100 simulations were run with different pedestrian and car paths. One of the things that were changed was the speed of vehicles, but it was always around 50 kmph. Additionally, the starting position of objects and the moment of entry of a pedestrian on the road were changed, as well as the speed. The acceleration during braking of the blue car was selected in such a way that it stopped at a given distance from pedestrians crossing the road. In addition, it was assumed that the dog was moving in front of the pedestrian at a slightly higher speed. Simulations were carried out for distances L1 of 2, 4, 6, 8, and 10 m. On the basis of simulation data, GSM was used to determine the probability of object detection using the functions shown in Figs. 4–7. To describe the probability of dog detection, data from the pedestrian class were used.

An example of the results obtained for the 6-m distance is shown in Fig. 12. As the graphs show, the dog was detected

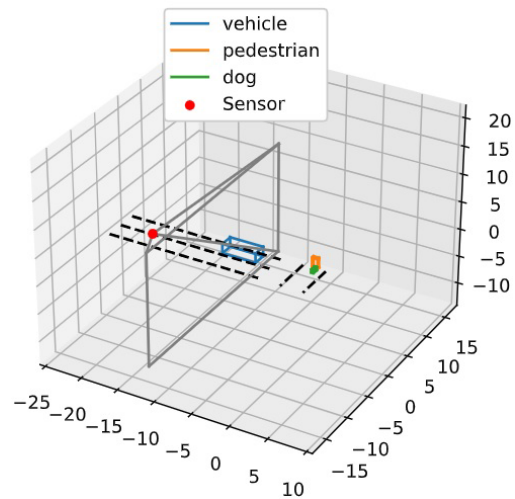


Fig. 11. Simulation frame.

faster than the pedestrian as it emerged from under the occluding car. In this case, the algorithms started detecting the pedestrian when the sensor was at a distance of about 13 m from the pedestrian and in the case of the dog about 17 m. Given that the braking distance of the vehicle is approximately 13–14 m, the implication is that assuming zero system response time, it is impossible to brake in front of the pedestrian if there were only that pedestrian in the zebra crossing. Moreover, the differences between the FCOS3D and Centerpoint algorithms are small and amount to less than a meter. The Centerpoint algorithm reacts slightly faster. However, these are not differences that would avoid a collision.

Table V presents distances to pedestrian for specific detection confidence levels and L1 distance of a stopped blue vehicle. The results obtained show that as soon as a blue vehicle stopped within 10 m of a pedestrian crossing, an object appeared in the field of vision of the sensors fast enough for the detection algorithms to detect it in time to brake in front of

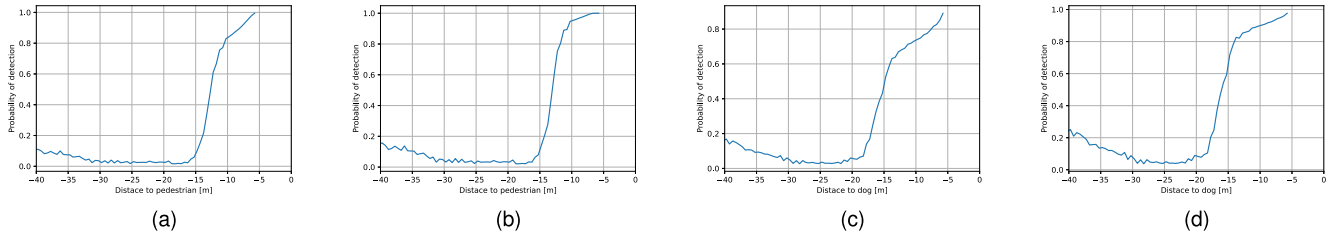


Fig. 12. Mean probability of object detection with respect to the distance to the object. (a) Pedestrian, FCOS3D. (b) Pedestrian, CenterPoint. (c) Dog, FCOS3D. (d) Dog, CenterPoint.

TABLE V

DISTANCES TO PEDESTRIAN FOR SPECIFIC DETECTION CONFIDENCE LEVELS AND L1 DISTANCE OF A STOPPED BLUE VEHICLE

Algorithm	L1 distance	Probability of detection		
		0.5	0.7	0.9
FCOS3D	2m	5.75m	4.75m	4.25m
	4m	8.25m	7.75m	7.25m
	6m	11.75m	10.25m	7.75m
	8m	14.75m	13.25m	7.75m
	10m	18.25m	13.75m	7.75m
Centerpoint	2m	5.75m	4.75m	4.25m
	4m	8.25m	7.75m	7.50m
	6m	12.25m	11.25m	10.25m
	8m	15.75m	14.25m	13.25m
	10m	19.25m	17.75m	14.25m

the pedestrian, assuming a legal speed of 50 kmph. Reducing the speed to 30 kmph and assuming a braking distance for this speed of about 6 m, the detection algorithms allow sufficient early detection when the L1 distance is 4 m. In this case, it is also noticeable that the lidar algorithm performs better and reacts slightly faster. To obtain a pedestrian detection probability of 0.9 using the FCOS3D algorithm, the sensor must be at a distance of about 8 m from the object in question.

The frequency of occurrence of the case study will depend on the volume of traffic on the road and the number of pedestrians trying to cross the carriageway. It should be noted that hypothetically a pedestrian, particularly a child, or an animal may intrude on the roadway at any point. The research carried out also generalizes to this case. It can occur on any multilane road when the preceding vehicle occludes an object that has encroached on the roadway.

VIII. CONCLUSION

On the basis of the conducted research, the following improvements of the transportation system, particularly guidelines for the reorganization of pedestrian crossing signage, traffic, measurement, and control systems, including algorithms with the prediction (risk assessment), are put forward.

- 1) For multilane pedestrian crossings, traffic lights shall be recommended. In the case of a lack of traffic lights:
 - a) Speed reduction toward 30 kmph.
 - b) Distance L1 is critical to pedestrian visibility and shall be set at least at 4 m for 30 kmph and 10 m for 50 kmph.
- 2) Perception systems must take into account that pedestrians or animals can also cross the road in unmarked places. Thus, advanced driver assistance systems should:
 - a) include a communication system between vehicles (V2V communication) to receive information on pedestrians at crossings more quickly;

- b) include detection algorithms that ought to track preceding cars that are in the lane next to them and be able to predict that when they brake, an object may unexpectedly emerge behind them; and
- c) include V2x communication (vehicle to everything), that is, a pedestrian crossing will detect when a pedestrian is at a crosswalk despite green lights for vehicles and inform oncoming traffic.

ACKNOWLEDGMENT

This work was carried out in cooperation with Aptiv Services Poland S.A.—Technical Center Kraków and AGH University of Science and Technology—Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, Department of Automatic Control and Robotics. This work was supported by the Polish Ministry of Science and Higher Education under Project 0014/DW/2018/02.

REFERENCES

- [1] H. A. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," 2018, *arXiv:1812.05784*.
- [2] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11784–11793.
- [3] T. Wang, X. Zhu, J. Pang, and D. Lin, "FCOS3D: Fully convolutional one-stage monocular 3D object detection," 2021, *arXiv:2104.10956*.
- [4] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [5] *2016 Fatal Motor Vehicle Crashes: Overview*, NHTSA's Nat. Center Statist. Anal., Nat. Highway Traffic Saf. Admin., Washington, DC, USA, 2017.
- [6] B. Schlager et al., "State-of-the-art sensor models for virtual testing of advanced driver assistance systems/autonomous driving functions," *SAE Int. J. Connected Automated Vehicles*, vol. 3, no. 3, pp. 233–261, Oct. 2020.
- [7] European Commission. *Intelligent Transport Systems: Road*. Accessed: Mar. 15, 2020. [Online]. Available: https://ec.europa.eu/transport/themes/its/road_en
- [8] H. Winner, S. Hakuli, F. Lotz, and C. Singer, *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*, 1st ed. Switzerland: Springer, 2015. [Online]. Available: <https://link.springer.com/referencework/10.1007/978-3-319-12352-3#bibliographic-information>
- [9] N. Kalra and S. M. Paddock, *Driving to Safety: How Many Miles of Driving Would it Take to Demonstrate Autonomous Vehicle Reliability?*. Santa Monica, CA, USA: RAND Corporation, 2016.
- [10] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. RA-2, no. 1, pp. 14–23, Mar. 1986.
- [11] T. Sulkowski, P. Bugiel, and J. Izydorczyk, "In search of the ultimate autonomous driving simulator," in *Proc. Int. Conf. Signals Electron. Syst. (ICSES)*, Sep. 2018, pp. 252–256.
- [12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.

- [13] PGAutomotive. *CarMaker*. Accessed: Jun. 15, 2021. [Online]. Available: <https://pg-automotive.com/products-services/simulation-software/carmaker/>
- [14] dSPACE. *ASM Traffic*. Accessed: Jun. 15, 2021. [Online]. Available: <https://www.dspace.com/en/pub/home.cfm>
- [15] M. Rozewicz and A. Pilat, "Modelling of bike steered by CMG," in *Proc. 21st Int. Conf. Methods Models Autom. Robot. (MMAR)*, Miedzyzdroje, Poland, Aug. 2016, pp. 595–600.
- [16] M. Rozewicz and A. Pilat, "Robust control of bicycle model with CMG," in *Proc. 21st Int. Conf. Methods Models Autom. Robot. (MMAR)*, Miedzyzdroje, Poland, Aug. 2016, pp. 369–374.
- [17] S. Muckenhuber, H. Holzer, J. Rubsam, and G. Stettinger, "Object-based sensor model for virtual testing of ADAS/AD functions," in *Proc. IEEE Int. Conf. Connected Vehicles Expo (ICCVE)*, Nov. 2019, pp. 1–6.
- [18] M. Stolz and G. Nestlinger, "Fast generic sensor models for testing highly automated vehicles in simulation," *e i Elektrotechnik und Informationstechnik*, vol. 135, nos. 4–5, pp. 365–369, Aug. 2018.
- [19] T. A. Wheeler, M. Holder, H. Winner, and J. M. Kochenderfer, "Deep stochastic radar models," 2017, *arXiv:1701.09180*.
- [20] N. Hirsenkorn, T. Hanke, A. Rauch, B. Dehlink, R. Rasshofer, and E. Biebl, "A non-parametric approach for modeling sensor behavior," in *Proc. 16th Int. Radar Symp. (IRS)*, Jun. 2015, pp. 131–136.
- [21] N. Hirsenkorn et al., "A ray launching approach for modeling an FMCW radar system," in *Proc. 18th Int. Radar Symp. (IRS)*, Jun. 2017, pp. 1–10.
- [22] F. Maier, V. Makkapati, and M. Horn, "Environment perception simulation for radar stimulation in automated driving function testing," *Elektrotechnik und Informationstechnik*, vol. 135, pp. 1–7, Aug. 2018.
- [23] K. Schuler, D. Becker, and W. Wiesbeck, "Extraction of virtual scattering centers of vehicles by ray-tracing simulations," *IEEE Trans. Antennas Propag.*, vol. 56, no. 11, pp. 3543–3551, Nov. 2008.
- [24] M. Jasinski, "A generic validation scheme for real-time capable automotive radar sensor models integrated into an autonomous driving simulator," in *Proc. 24th Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2019, pp. 612–617.
- [25] S. Schmidt, B. Schlager, S. Muckenhuber, and R. Stark, "Configurable sensor model architecture for the development of automated driving systems," *Sensors*, vol. 21, no. 14, p. 4687, Jul. 2021.
- [26] T. Hanke, N. Hirsenkorn, B. Dehlink, A. Rauch, R. Rasshofer, and E. Biebl, "Generic architecture for simulation of ADAS sensors," in *Proc. 16th Int. Radar Symp. (IRS)*, Jun. 2015, pp. 125–130.
- [27] H. Li et al., "Phenomenological modelling of camera performance for road marking detection," *Energies*, vol. 15, no. 1, p. 194, Dec. 2021.
- [28] K. Lelowicz, M. Jasinski, and M. Piatek, *Generic Sensor Model for Object Detection Algorithms Validation*. Switzerland: Springer, 2020, pp. 1249–1260.
- [29] C. Bustos et al., "Explainable, automated urban interventions to improve pedestrian and vehicle safety," *Transp. Res. C, Emerg. Technol.*, vol. 125, Apr. 2021, Art. no. 103018.
- [30] A. Kalatian and B. Farooq, "A context-aware pedestrian trajectory prediction framework for automated vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 134, Jan. 2022, Art. no. 103453.
- [31] H. Razali, T. Mordan, and A. Alahi, "Pedestrian intention prediction: A convolutional bottom-up multi-task approach," *Transp. Res. C, Emerg. Technol.*, vol. 130, Sep. 2021, Art. no. 103259.
- [32] *Dziennik Ustaw Rzeczypospolitej Polskiej, Warszawa, 26 listopada 2019 r., Poz. 2310, Rozporządzenie Ministrów Infrastruktury oraz Spraw Wewnętrznych i Administracji z dnia 31 lipca 2002 r. w sprawie znaków i sygnałów drogowych*, Warsaw, Poland. [Online]. Available: <https://isap.sejm.gov.pl/isap.nsf/download.xsp/WDU20190002310/O/D20192310.pdf>
- [33] (1997). U.K. Statutory Instruments. *The Zebra, Pelican and Puffin Pedestrian Crossings Regulations and General Directions 1997*. Statutory Instruments. [Online]. Available: <https://www.legislation.gov.uk/uksi/1997/2400/introduction/made>
- [34] H. J. Escalante and K. Hofmann, "NeurIPS 2020 competition and demonstration track: Revised selected papers," in *Proc. NIPS Competition Demonstration Track*, vol. 133, H. J. Escalante and K. Hofmann, Eds., Dec. 2021, pp. 1–2.
- [35] (2020). MMDetection3D Contributors. *MMDetection3D: OpenMMLab Next-Generation Platform for General 3D Object Detection*. [Online]. Available: <https://github.com/open-mmlab/mmdetection3d>
- [36] H. Caesar et al., "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11621–11631.
- [37] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.
- [38] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12689–12697.
- [39] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.



Kamil Lelowicz (Member, IEEE) received the master's degree in automatic control and robotics from the AGH University of Science and Technology, Kraków, Poland, in 2018, where he is currently pursuing the Ph.D. degree.

He is also engaged in commercial research and development and product development activities of advanced automotive driver assistance systems and cabin monitoring systems. His research interests include modeling and simulation, computer vision, and machine learning.

Mr. Lelowicz is a member of the Polish Chapter of the IEEE Vehicular Technology Society.



Adam Krzysztof Piłat (Member, IEEE) received the M.Sc., Ph.D., and D.Sc. degrees in automatic control and robotics from the AGH University of Science and Technology, Kraków, Poland, in 1996, 2002, and 2014, respectively.

He is a Professor with the AGH University of Science and Technology. He is currently employed at the Department of Automatic Control and Robotics as a Researcher, a Didactic, and the Head of the Photovoltaic, Robotics and Magnetic Levitation Laboratory. He has authored

or coauthored 44 articles, 85 papers, four books, 12 book chapters, and 20 patents. His scientific interest is focused on interdisciplinary modeling, simulation, controller synthesis, and real-time control. His research is mainly focused on device design and prototyping with embedded active magnetic levitation technology.

Dr. Piłat is a member of the IEEE Control Systems Society and the Robotics and Automation Society.