IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 December 2023, accepted 24 December 2023, date of publication 11 January 2024,
date of current version 19 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3353051

RESEARCH ARTICLE

3D Object Detection for Self-Driving Vehicles Enhanced by Object Velocity

LEANDRO ALEXANDRINO[1,2,3], HADI Z. OLYAEI[3], ANDRÉ ALBUQUERQUE[3], PETIA GEORGIEVA[1,2,4], (Senior Member, IEEE), AND MIGUEL V. DRUMMOND[1,2], (Member, IEEE)

[1]Instituto de Telecomunicações, 3810-193 Aveiro, Portugal
[2]Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, 3810-193 Aveiro, Portugal
[3]Bosch Car Multimedia, 4705-820 Braga, Portugal
[4]Institute of Electronics and Informatics Engineering of Aveiro (IEETA), University of Aveiro, 3810-193 Aveiro, Portugal

Corresponding author: Leandro Alexandrino (l.alexandrino@ua.pt)

This work was supported in part by the European Union (EU)-NextGenerationEU through the National Recovery and Resilience Plan, Bulgaria, under Project BG-RRP-2.004-0005; and in part by Fundação para a Ciência e Tecnologia/Ministério da Ciência, Tecnologia e Ensino Superior (FCT/MCTES) through the National Funds co-funded EU Funds under Project UIDB/50008/2020-UIDP/50008/2020.

ABSTRACT A large number of vision sensors has been proposed for enabling self-driving vehicles to perceive their surroundings. Among them, Light Detection And Ranging (LiDAR) presents the unique advantage of acquiring a high resolution 3D representation of the vehicle surroundings, in the form of point clouds, which enables accurate 3D object detection. The success of the first (and current) generation LiDARs has motivated the development of a second generation of this sensor, now based on coherent detection. Second generation LiDARs thus enable not only estimating radial distance, but also radial velocity for each point of the point cloud. The objective of this work is to investigate which benefits can be obtained by considering such an additional information – radial velocity – in 3D object detection. Results show that considering object velocity is particularly helpful in objects represented by a small number of points.

INDEX TERMS Autonomous driving, 3D object detection, coherent LiDAR, point cloud, radial velocity, deep learning.

I. INTRODUCTION

The number of road traffic accidents remains tragically high [1]. Such accidents frequently result in fatalities or lifetime disabilities for the victims, being a worldwide health concern. The main cause of such unfortunate reality is human error [2]. Despite the availability of safety systems in modern vehicles, reckless driving is the main factor behind human error. Autonomous vehicles are the solution for such problems, as reckless driving becomes impossible.

In order to ensure safe driving, self-driving vehicles must perceive the surrounding environment with detail. This is possible using vision sensors. Ultrasonic, Camera, Radio Detection And Ranging (RaDAR) and Light Detection And Ranging (LiDAR) are among the commonly proposed vision sensors.

From the mix of different sensors that have been proposed, LiDAR is emerging as a key vision sensor for vehicle perception tasks due to its ability to acquire a precise 3D representation of the surroundings [3], [4], [5]. Driven by the success of the LiDAR in critical perception tasks, such as 3D object detection, the industry and the research community have been working on a second generation LiDAR, typically resorting to coherent detection. This sensor enables the estimation of the object distance with the relative radial velocity also being directly acquired per-point. Since this additional feature is acquired without a significant overhead, there is an interest in using this information for perception tasks, such as 3D object detection. Object velocity can help in identifying objects moving at different speeds. For instance, if the vehicle is moving, stationary background features can

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda.

be easily filtered out. Moreover, road objects such as car, bicycles and pedestrians typically have different velocities. Thus, velocity information can help in better classifying and segmenting different objects.

This paper investigates the benefits brought by the use of the radial velocity additional feature in LiDAR-based 3D object detection using Deep-Learning (DL). To the best of our knowledge, such an investigation has not been done before.

This paper is organized in 9 sections. Sections I presents an introduction and overview of self-driving vehicles and widely adopted vision sensors, such as LiDAR. Section II summarises 3D object detection tools for processing first and second-generation LiDAR point clouds. Sections III and IV present the proposed approaches to generate a synthetic dataset and to validate it, respectively. Section V describes the methodology applied for DL-based 3D object detection. Section VI discusses the results for the defined use cases. Section VII presents the conclusions with respect to the use cases. Finally, Section VIII summarises the major contributions of this study.

II. CONTRIBUTIONS

The main contributions of this work are:

- The first publicly available LiDAR dataset including the radial velocity. Available in [25];
- An improved dataset generation tool, with enhanced labelling consistency and with the acquisition of five different point clouds. The code is available in [21];
- The first study regarding the benefits brought by the radial velocity as an additional feature in the LiDAR-based 3D object detection with Deep Learning models.

With the public availability of such code, we enable the community to work on 3D object detection for automotive scenarios using synthetic dataset generation.

III. 3D OBJECT DETECTION USING LIDAR POINT CLOUDS

The first (and current) generation of automotive LiDAR sensors is based on time of flight estimation. Each point of the acquired point cloud is typically comprised by the cartesian coordinates (x, y, z) and the reflection intensity, i.

Motivated by the success and availability of the first-generation LiDAR sensors, contributions have been made over the recent years for 3D object detection:

- **Toolboxes**, such as OpenPCDet [6], that provide a user-friendly interface for experimenting with different 3D object detection models and datasets;
- **3D object detection models**, such as SECOND [7], PointRCNN [8], PointPillars [9], among others. PointPillars was the chosen 3D Object Detection model for this work due to its simplicity and robustness, providing a good trade-off between model performance and training time, while also being available in OpenPCDet;
- **Real-world datasets**, such as Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) [10], Waymo [11], nuScenes [12], among others. KITTI is the most popular real-world dataset due to being the first annotated dataset made publicly available for mobile robotics and autonomous driving purposes;
- **Benchmarks**, such as KITTI benchmark [13]. Due to the popularity of the dataset, the KITTI metrics are also commonly adopted by the research community for 3D object detection tasks.

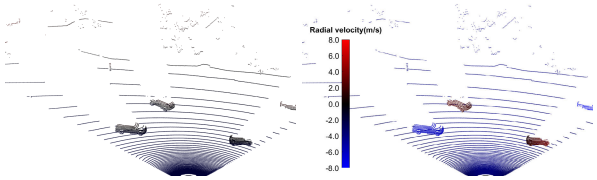


FIGURE 1. Left: first-generation LiDAR point cloud. Right: second-generation LiDAR point cloud. Point clouds colored by relative radial velocity.

A second-generation LiDAR typically resorts to coherent detection, with Frequency-Modulated Continuous-Wave (FMCW) being the most popular signaling method. Distance and radial velocity can be directly estimated from the positive and the negative linear chirps of the received FMCW waveform [14]. As seen in Figure 1, each point of the second-generation LiDAR point cloud is typically comprised by the cartesian coordinates (x, y, z), the intensity, i, and the radial velocity, v_r.

The second-generation LiDAR is still under development. Scantinel Photonics [15], Aurora [16], Aeva [17] and Baraja [18] are among the most popular coherent LiDAR sensor development companies, with prototypes, demonstrations and collaborations with automotive manufacturers. Since the majority of coherent LiDAR sensors are prototypes it is natural that, to the best of the authors' knowledge, there are no coherent LiDAR datasets made publicly available. Consequently, the impact of the additional feature – radial velocity – for LiDAR-based Deep-Learning based 3D object detection is yet to be assessed. The study in [19] tried to perform such an analysis. However, since the real-world acquired dataset was comprised by a small number of frames, no Deep learning 3D object detection models were used. Furthermore, such dataset was not made publicly available. The present work addresses both gaps: we generate and make publicly available a dataset made of point clouds obtained by an ideal coherent LiDAR, and investigate which benefits can be obtained from using the radial velocity as an additional feature in 3D object detection using Deep Learning models. In order to perform such investigation, we resort to the existent tools for first-generation LiDAR sensors, such as OpenPCDet, PointPillars and KITTI benchmark, and generate a synthetic dataset with a KITTI-like format to ensure that the generated dataset can be directly processed by the toolbox.

IV. SYNTHETIC DATASET GENERATION

The generation of a real-world dataset made of point clouds captured by a second generation LiDAR is currently not a

TABLE 1. Object type labels difference between KITTI and PreSIL.

Object type	KITTI	PreSIL
Airplane	✗	✓
Boat	✗	✓
Bus	✗	✓
Car	✓	✓
Cyclist	✓	✓
DontCare	✓	✗
Industrial	✗	✓
Misc	✓	✗
Motorbike	✗	✓
Pedestrian	✓	✓
Person_sitting	✓	✓
Railed	✗	✓
Trailer	✗	✓
Tram	✓	✗
Truck	✓	✓
Utility	✗	✓
Van	✓	✗

TABLE 2. KITTI occlusion annotation. Reproduced from [23].

Occlusion value	Occlusion state
0	fully visible
1	partly occluded
2	largely occluded
3	unknown

viable option, given that there are only protoypes of such LiDARs. We therefore had no other option than to generate a synthetic dataset. The Grand Theft Auto V (GTA V) video-game was selected as the source of the data, due to the realistic lifelike scenarios and the availability of native functions that return in-game object information. Precise Synthetic Image and LiDAR (PreSIL) [20] was used as the dataset generator since it provides improvements over the native ray-casting functions and object representations [20]. Furthermore, this generator acquires a dataset with KITTI-like format, thus, the data can be directly processed by existent toolboxes, such as OpenPCDet.

The point clouds of the generated dataset do not have intensity information, as such information is not provided by GTA V. Thus, the generated point clouds were not 5D (i.e., (x, y, z, i, v_r)), but 4D (i.e., (x, y, z, v_r)), as are the point clouds of the KITTI dataset, (x, y, z, i). This means that no additional complexity was added to the considered PointPillars model. Furthermore, previous tests [24] have shown that processing the KITTI dataset while discarding intensity information negligibly degraded performance. In summary, in comparison to KITTI point clouds, and to point clouds obtained by most mainstream LiDARs, the same 3D object detection models can be used by replacing intensity information with radial velocity information without risking losing performance.

FIGURE 2. Vehicle models used as reference for new object type annotation.

Although the PreSIL dataset generator was fairly mature, modifications were done to enhance the dataset quality and consistency. Furthermore, new features were implemented to acquire the required point clouds for this study. The updated version of the dataset generator is accessible in [21].

A. IMPROVING ANNOTATION QUALITY

Some annotations computed by the original PreSIL generator were not consistent with the KITTI labelling format. Such inconsistency would deteriorate the model performance. Therefore, modifications were made in the object type and occlusion annotations.

1) OBJECT TYPE ANNOTATION

As shown in Table 1, PreSIL object type annotations differ from the KITTI dataset. In order to enhance consistency between the real-world dataset and the synthetic dataset, the object types were changed to match the KITTI dataset.

Additionally, it was noticed that some objects from the same vehicle segment, which are commonly present in the video-game scenarios, were incorrectly labelled, such as the 'Cavalcade' GTA V vehicle model which was labelled by PreSIL as a 'Truck' type object. In order to improve the consistency and quality of the dataset, the vehicle model to object type relation of each vehicle model was reviewed, as depicted in Figure 2, with:

- Small-size vehicles are considered 'Car' for labelling purposes. Cavalcade vehicle was used as reference model;
- Medium-size vehicles are considered 'Van' for labelling purposes. Taco vehicle was used as reference model;
- Large-size vehicles are considered 'Truck' for labelling purposes. Benson vehicle was used as reference model.

2) OBJECT OCCLUSION ANNOTATION

The KITTI object occlusion label is an integer value, as shown in Table 2. Depending on the object truncation, occlusion and bounding box height, an object can be assigned a Easy, Moderate or Hard difficulty level [22].

Nonetheless, PreSIL yields a float value from the occlusion computation. Thus, incorrect difficulty levels can be assigned to objects. Since KITTI provides no threshold for such attribution, a visual inspection was performed to establish the following thresholds:

- If the computed PreSIL occlusion is lower than 0.2, the object has the KITTI occlusion level 0;

- If the computed PreSIL occlusion is greater than or equal to 0.2 while being lower than 0.6, the object has the KITTI occlusion level 1;
- If the computed PreSIL occlusion is greater than or equal to 0.6, the object has the KITTI occlusion level 2;
- No level 3 occlusion exists since the occlusion is computed for every object.

B. IMPROVING FRAME DIVERSITY

When capturing frames (i.e., images and point clouds) at a high frame rate, consecutive frames have little to no difference among them. As such, one could be biasing the model with identical frames in the training and testing sets. Conversely, using a low frame rate significantly increases the duration involved in acquiring a dataset. Therefore, our approach used 0.2 Hz as the frame rate specification with the following approaches to mitigate frame similarity:

- Reduce the frame rate on the dataset acquisition when the vehicle is in the same position in consecutive frames. It was noticed that consecutive frames present high resemblance when vehicles are stopped awaiting traffic light changes. When the vehicle is moving, the frame rate is restored to the original specification;
- Frame shuffling was performed after the dataset acquisition to select which frames comprise the dataset. In order to generate a dataset with 15000 frames, approximately the same as KITTI, over 25000 frames that had annotated objects were acquired. From the 25000 frames, a random shuffle was performed to select which frames comprise the final dataset;
- Random frame shuffling for the testing and training splits was performed in the final dataset to further enhance the diversity of frames.

C. ONLINE POINT CLOUDS

The default PreSIL dataset generator source code acquires a single point cloud comprised by the object cartesian coordinates, (x, y, z), and by the object unique identifier, *ObjectID*. Since the point clouds required for this study could not be captured online by default, the acquisition of the following point clouds were included in the improved dataset generator:

1) **Point cloud 1:** $(x, y, z$**,(Bool)Is_Object**$)$. The best performance of the Deep Learning model is expected for such a point cloud as ground truth information is provided as the additional feature. This point cloud is acquired through the instance segmentation available in the GTA V game-engine, where each object is uniquely identified. Since two predominant object categories are commonly observed in the video-game environment, this point cloud is further divided into two point clouds.
2) **Point cloud 1A:** $(x, y, z$**,(Bool)Is_Car**$)$. The additional feature of each point that belongs to an object of the 'Car' type has a Boolean 1.0 value; contrariwise, the 0.0 value was used. An example is shown in Figure 3a.
3) **Point cloud 1B:** $(x, y, z$**,(Bool)Is_Ped**$)$. The additional feature of each point that belongs to an object of the 'Pedestrian' type has a Boolean 1.0 value; contrariwise, the value 0.0 was used. An example is shown in Figure 3b.
4) **Point cloud 2:** $(x, y, z$**,(Float)Radial_Velocity**$)$. Similarly to the data yielded by a coherent LiDAR sensor, this point cloud includes the relative radial velocity as the additional feature of each point, as shown in Figure 3c. Since no native function is available to acquire the radial velocity, it can be computed as:

$$v_{\mathrm{r}} = ||\vec{v_{\mathrm{TS}}}|| \cos(\theta) = \vec{v_{\mathrm{TS}}} \cdot \frac{\vec{r_{\mathrm{TS}}}}{||\vec{r_{\mathrm{TS}}}||}, \tag{1}$$

where:

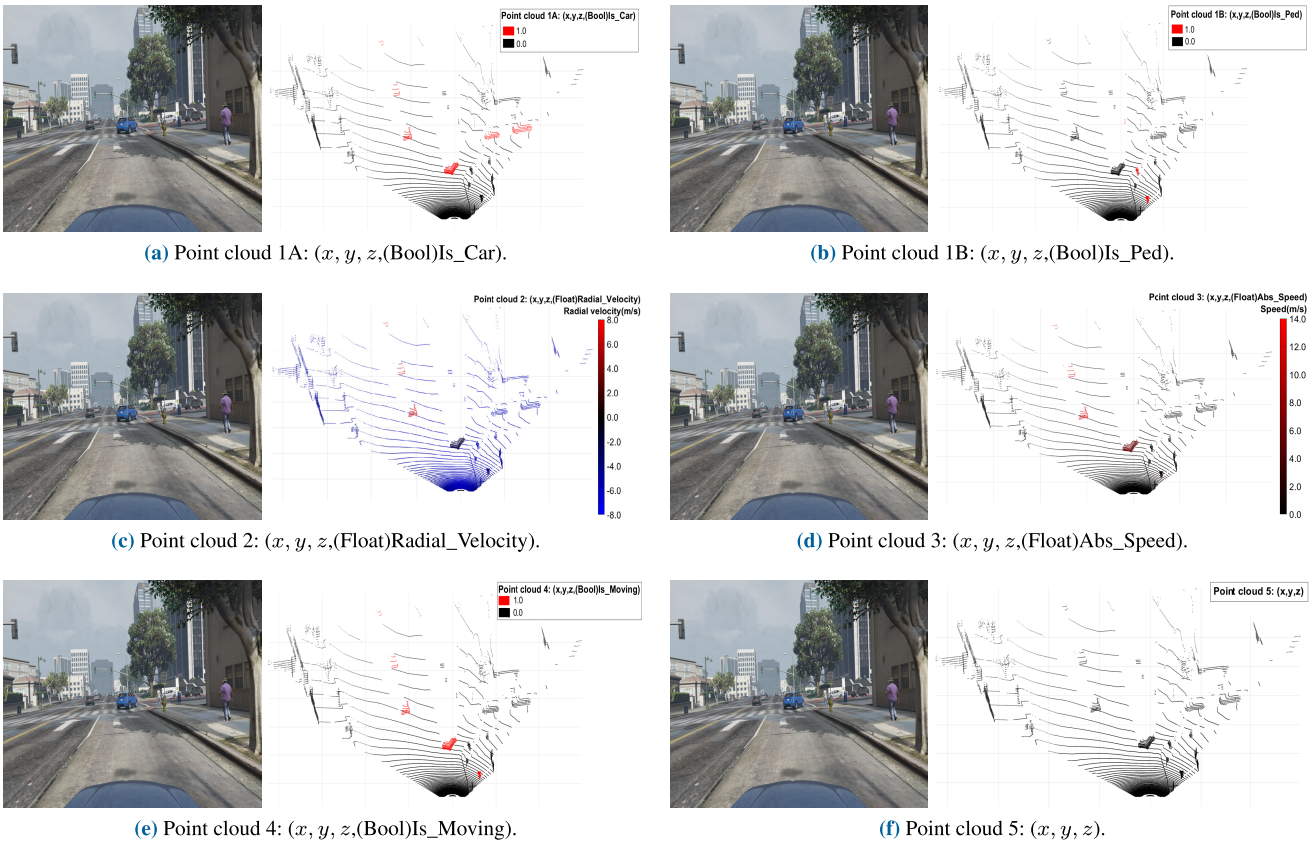
- v_{r} is the relative radial velocity and the additional feature appended to the point cloud;
- θ is the angle between the object velocity vector and the line of sight of the ego-vehicle;
- $\vec{v_{\mathrm{TS}}}$ is the relative velocity vector, computed by $\vec{v_{\mathrm{TS}}} = \vec{v_{\mathrm{T}}} - \vec{v_{\mathrm{S}}}$. The ego-vehicle velocity, $\vec{v_{\mathrm{S}}}$, and the object velocity vector, ($\vec{v_{\mathrm{T}}}$, were acquired through native functions;
- $\vec{r_{\mathrm{TS}}}$ is the relative position vector, computed by $\vec{r_{\mathrm{TS}}} = \vec{r_{\mathrm{T}}} - \vec{r_{\mathrm{S}}}$. The target point position, $\vec{r_{\mathrm{T}}}$, and the ego-vehicle position vector, $\vec{r_{\mathrm{S}}}$, can be acquired through a native function that returns GTA V world coordinates. However, in order to correctly resemble KITTI LiDAR sensor placement, the ego-vehicle position z-axis value was modified to 1.73 meters above ground, thus $\vec{r_{\mathrm{S}}} = (r_{s_x}, r_{s_y}, \mathrm{Ground} + 1.73)$.

For further details please check section 3.2.2 of [24].

5) **Point cloud 3:** $(x, y, z$**,(Float)Abs_Speed**$)$. In this point cloud, every point has the absolute speed of the object as the additional feature, yielded by a native function. An example is shown in Figure 3d.
6) **Point cloud 4:** $(x, y, z$**,(Bool)Is_Moving**$)$. In this point cloud, the additional feature is a Boolean value that is set to 1.0 if the object is moving; contrariwise it is set to 0.0 for static objects. An example is shown in Figure 3e.
7) **Point cloud 5:** (x, y, z). The worst performance of the Deep Learning model is expected since no additional information is provided. Therefore, the object detection model must only resort to the geometrical information of the objects for the predictions. An example is shown in Figure 3f.

D. NOISE ADDITION

Having a direct estimation of the radial velocity helps in distinguishing moving objects from stationary background points (e.g., ground points). It is therefore expected that having a direct estimation of the radial velocity helps relaxing the precision of range estimation. In order to evaluate such proposition, a new point cloud was created offline from both point clouds 2 and 5. For both point clouds, the point



(a) Point cloud 1A: $(x, y, z,$(Bool)Is_Car).

(b) Point cloud 1B: $(x, y, z,$(Bool)Is_Ped).

(c) Point cloud 2: $(x, y, z,$(Float)Radial_Velocity).

(d) Point cloud 3: $(x, y, z,$(Float)Abs_Speed).

(e) Point cloud 4: $(x, y, z,$(Bool)Is_Moving).

(f) Point cloud 5: (x, y, z).

FIGURE 3. Example of the acquired point clouds. Left: image of an acquired frame. Right: point cloud colored by the additional feature for the acquired frame.

TABLE 3. Object occurrences in the GTA V synthetic dataset.

Object type	#Occurrences
Car	56372
Pedestrian	23636
Misc	7505
Truck	4694
Van	1925
Person_sitting	590
Cyclist	57
DontCare	0
Tram	0

TABLE 4. Moving object occurrences with the average speed of the captured object types.

Object type	Moving	Average speed (km/h)
Car	74%	39
Pedestrian	47%	5
Misc	67%	27
Truck	84%	35
Van	82%	38
Person_sitting	0%	0
Cyclist	65%	22

coordinates (x, y, z) are affected by errors. For a matter of simplicity, no error was added to the radial velocity.

As observed in Figure 4, noise was added to the point cloud after its acquisition (thus, offline) by:

1) Converting the cartesian coordinates to spherical coordinates;
2) Generating an error using a gaussian distribution with the ideal radial distance as mean and standard deviation of 5 cm, 10 cm and 15 cm;
3) Converting the spherical coordinates back to cartesian coordinates.

V. SYNTHETIC GTA V DATASET VALIDATION

Generating a dataset, by itself, does not guarantee its quality. As no GTA V scenarios were user-customized, the game-engine is responsible for the location and density of all objects. The analysis of the object type occurrences, the object difficulty levels and the velocity occurrences follow what would be expected in everyday life scenarios, as:

- 'Car' and 'Pedestrian' type objects are the two predominant object classes, as shown in Table 3.
- There are a reasonable number of occurrences of each KITTI difficulty level objects (Easy, Moderate and Hard) in the dataset, as shown in Figure 5.

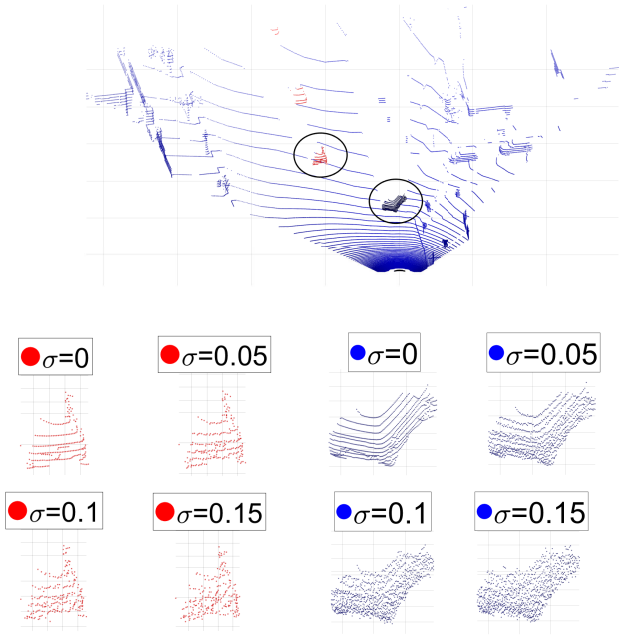


FIGURE 4. Example of the noise included in the point clouds. Top: A point cloud with two selected objects. Red: The approaching object points with different Gaussian error standard deviations (in meters). Blue: The receding object points with different Gaussian error standard deviations (in meters).

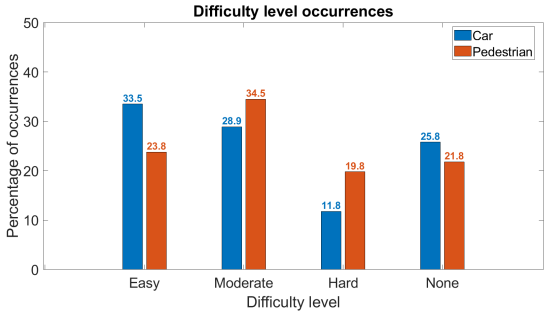


FIGURE 5. Difficulty level occurrences for 'Pedestrian' and 'Car' objects.

- The object type speeds analysis indicates that, on average, objects can be distinguished by their speed and there are significant occurrences of moving and static objects, as shown in Table 4.

VI. OBJECT DETECTION METHODOLOGY

The generated dataset is comprised by 15000 frames and three use cases were created:

1) Use case 1, with raw radial velocity as additional feature (as seen in Point cloud 2);
2) Use case 2, with processed radial velocity as additional feature (as seen in Point cloud 3 and 4);
3) Use case 3, with raw radial velocity as additional feature with erroneous point coordinates;

Four-fold cross-validation was applied for Use case 1. For the remaining use cases, a single fold of test and training split was applied. Each fold is comprised by 3750 randomly

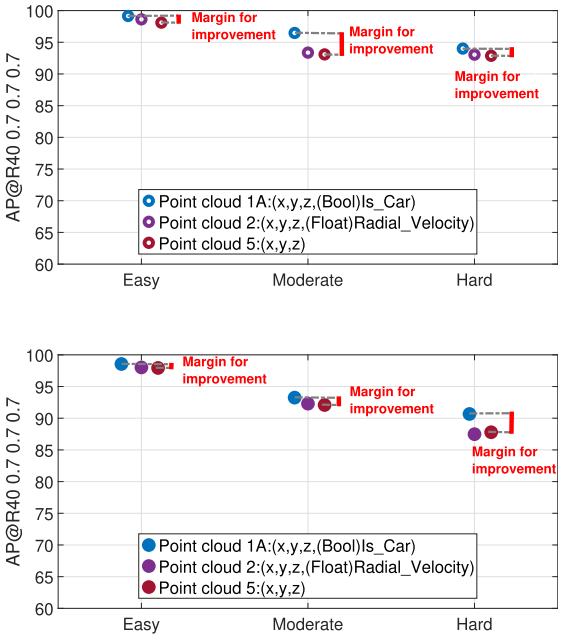


FIGURE 6. 3D AP@R40 'Car' type objects results using the radial velocity additional feature. Top: Training results. Bottom: Testing results.

assigned frames. For each iteration 11250 frames are used for model training, and 3750 frames for model testing.

The metrics employed by the KITTI benchmark for 3D object detection were adopted. The models are evaluated using the Average Precision (AP) metric, with 40-point interpolation, for three KITTI difficulty levels: Easy, Moderate and Hard.

The PointPillars model hyper-parameters were set at the default values of OpenPCDet for the KITTI dataset. The intensity information provided by the KITTI dataset was replaced with the additional feature (the velocity) of each acquired synthetic point cloud. For use case 1, the model was trained with 300 epochs, whereas for the remaining use cases 80 epochs were used.

VII. RESULTS

A. USE CASE 1: IMPACT OF RAW RADIAL VELOCITY

To analyse the impact of the raw radial velocity use case 1 was created. It is comprised by:

- **Upper limit**: established by point cloud 1A: $(x, y, z,$ (Bool)Is_Car) for 'Car' type objects or the point cloud 1B: $(x, y, z,$(Bool)Is_Ped) for 'Pedestrian' type objects. This is the upper limit as the additional feature is the ground truth;
- **Lower limit**: established by the point cloud 5:(x, y, z). This is the lower limit because there is no additional feature;
- **Margins for improvement**: obtained by the AP difference between the upper limit and the lower limit;
- The point cloud with the **relative radial velocity** additional feature, point cloud 2: $(x, y, z,$(Float) Radial_Velocity)

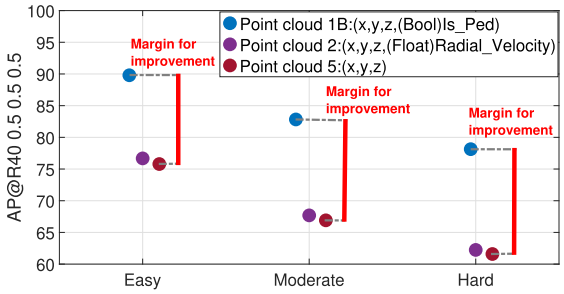


FIGURE 7. Use case 1 3D AP@R40 testing results for 'Pedestrian' type objects using the raw radial velocity additional feature.

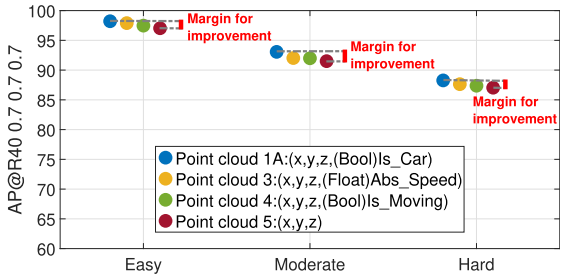


FIGURE 8. Use case 2 3D AP@R40 testing results for 'Car' type objects using speed and motion information.

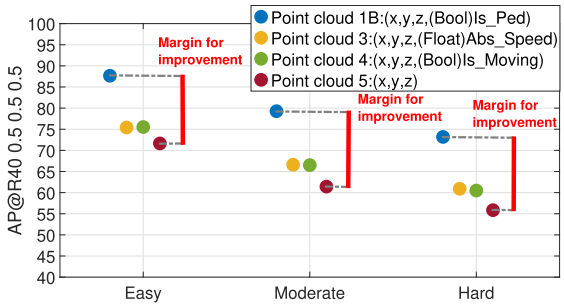


FIGURE 9. Use case 2 3D AP@R40 testing results for 'Pedestrian' type objects using speed and motion information.

1) 'CAR' TYPE OBJECTS

The AP averaged from all four folds, for 'Car' type objects, are shown in Figure 6.

The results indicate that there is no overfitting, since there is no substantial difference between the training and the testing performance. Overfitting was not observed in any of the results, and therefore we only show test performances henceforth.

Analysing the lower limit, though no additional feature is used, excellent performance was nonetheless achieved. Such a good performance was achieved likely because the number of points that comprise a Car is large. Therefore, objects which are comprised by a large number of points have, necessarily, a richer geometrical representation of their shape and structure, thus not requiring additional information to be detected. Since the geometrical information of the cars is sufficient by itself to achieve such a high performance, the obtained margin for improvement was small, of at most 3 %. Therefore, the radial velocity information introduced, at best, slight improvements for Easy and Moderate difficulty levels objects. Moreover, on average, the model performance at the Hard difficulty level slightly decreased with the use of this feature.

2) 'PEDESTRIAN' TYPE OBJECTS

A similar analysis was performed for 'Pedestrian' type objects. The average fold AP results are shown in Figure 7.

For 'Pedestrian' type objects there is a significant margin for improvement, with approximately 15 % improvement potential for all difficulty levels. Following the rationale presented when analysing 'Car' type objects, this occurs because 'Pedestrian' type objects are typically represented by a small number of points, which makes such objects harder to detect. Nonetheless, despite existing a large margin for improvement, the obtained improvements were negligible ($\sim$ 1 %).

B. USE CASE 2: IMPACT OF PROCESSED RADIAL VELOCITY

Since the benefits obtained from the raw radial velocity were minimal, perhaps processing this feature into simpler features yields better results. As such, use case 2 was created. It is comprised by:

- **Upper limit**: established by point cloud 1A: (x, y, z, (Bool)Is_Car) for 'Car' type objects or the point cloud 1B: (x, y, z,(Bool)Is_Ped) for 'Pedestrian' type objects. This is the upper limit as the additional feature is the ground truth;
- **Lower limit**: established by the point cloud 5:(x, y, z). This is the lower limit because there is no additional feature;
- **Margins for improvement**: obtained by the AP difference between the upper limit and the lower limit;
- The point cloud with the **absolute ground speed** additional feature per-point, point cloud 3: (x, y, z, (Float)Abs_Speed);
- The point cloud with a **Boolean value for moving objects** additional feature per-point, point cloud 4: (x, y, z,(Bool)Is_Moving);

1) 'CAR' TYPE OBJECTS

The use case 2 results for 'Car' type objects is presented in Figure 8.

The results follow the same tendency observed for use case 1. The margin for improvement is minimal for 'Car' type objects and no significant benefits were obtained from the processed radial velocity additional feature.

2) 'PEDESTRIAN' TYPE OBJECTS

The use case 2 results for 'Pedestrian' type objects is presented in Figure 9.

There was a slight increase in the margins for improvement when compared to the use case 1, caused by the different model training conditions, with a margin of 16 % to 18 %. Analysing the impact of the processed radial velocity, it can be seen that both the processed features were able to

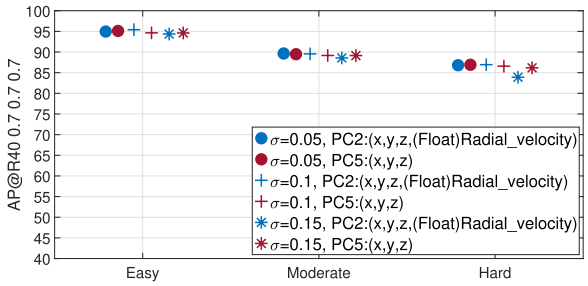


FIGURE 10. 3D AP@R40 'Car' type object testing results for the use case 3. PC used as an abbreviation of point cloud.

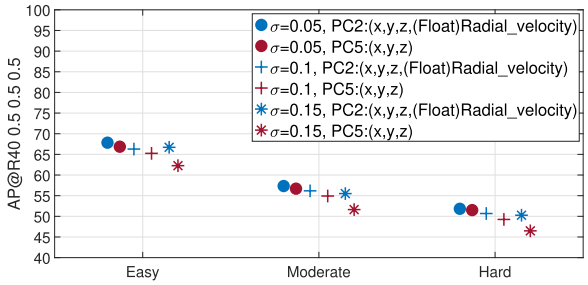


FIGURE 11. 3D AP@R40 'Pedestrian' type object testing results for the use case 3. PC used as an abbreviation of point cloud.

introduce more benefits to the model performance, with up to approximately 5 %. A possible explanation for such an improvement is that the model prefers just knowing which points are moving, and which points are not, as such a binary information resembles ground truth.

C. USE CASE 3: IMPACT OF RAW RADIAL VELOCITY FOR ERRONEOUS RANGE ESTIMATION

So far, point clouds with ideal range estimation were employed. In order to observe whether considering ideal radial velocity could help in better detecting objects in points clouds with non-ideal range estimation, use case 3 was created. It is comprised by:

- **Lower limit**: achieved by using the point cloud 5: $(x, y, z) + (\epsilon_x,\epsilon_y,\epsilon_z)$. This point cloud includes error in the coordinates of each point.
- The point cloud with the **radial velocity**, point cloud 2: $(x, y, z,$(Float)Radial_velocity$) + (\epsilon_x,\epsilon_y,\epsilon_z,0)$, containing the same **coordinates error** as in the other point cloud that comprises this use case. The raw radial velocity additional feature remains ideal;

1) 'CAR' TYPE OBJECTS

The results for 'Car' type objects are shown in Figure 10.

Figure 10 shows that, despite the slight performance decrease obtained from degrading the geometric information quality, the velocity information did not improve the object detection. In fact, it slightly decreased the model performance when compared to the scenario without the additional feature and the coordinates being affected by the highest introduced error (noise). The raw radial velocity did not introduce any noteworthy benefits to this use case.

2) 'PEDESTRIAN' TYPE OBJECTS

The results of use case 3 for 'Pedestrian' type objects are presented in Figure 11.

Note that, as the average error introduced to the object coordinates increases, the model was able to extract more benefits from the radial velocity information due to the poor geometrical information of the 'Pedestrian' type objects. Hence, the raw radial velocity was able to provide benefits (approximately 5 % improvement) to such a challenging and critical task.

Raw radial velocity already proved not to help neither in localizing nor in classifying objects with many points, such as cars. Hence, feeding the model with an extra feature that is not useful for detecting such an object class either makes no difference in its performance, or may even be detrimental if data is noisy, as is the case of Figure 11. Conversely, for pedestrians, Figure 10, this is not the case, as pedestrians comprise few points. Hence, even if data is noisy, considering radial velocity helps in detecting such objects.

VIII. CONCLUSION

In this paper, we investigated how the per-point radial velocity information provided by an ideal second-generation LiDAR would benefit 3D object detection in automotive scenarios. In general, having radial velocity information available for each point of the point cloud helps in differentiating moving objects from stationary background points. Such help is particularly valuable when the object is small and/or comprises a small number of points. As a result, radial velocity helps where it is most critical: in detecting objects represented by a small number of points, such as pedestrians.

Three scenarios were studied to support such conclusions. In the first scenario the raw radial velocity was considered as an additional feature. In the second scenario, the radial velocity was processed into a simpler feature, such as a Boolean value for moving objects or object absolute speed. Finally, in the third scenario, we tested the premise of whether ideally estimated radial velocity would help detecting objects, provided that the cartesian coordinates (x, y, z) are non-ideally estimated, as it occurs in practice.

The results obtained for 'Car' type objects in all three scenarios showed that the radial velocity, either raw or processed, produced negligible benefits. The reason behind such a negligible improvement lies in the large number of points that represent a car, which enable accurately detecting a Car without requiring further information.

Conversely, for 'Pedestrian' type objects, a significant margin for improvement, up to 15 %, was estimated. Such makes sense, as pedestrians are represented by a small number of points. Considering the raw radial velocity introduced negligible minimal improvements, approximately 1 %, and 5 % for non-ideal points clouds with error in range estimation. However, the simpler features acquired

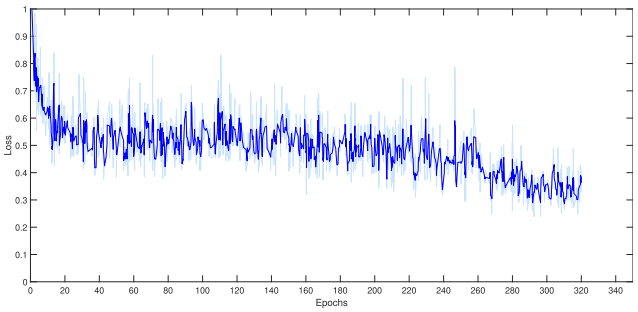


FIGURE 12. Loss function using point cloud 5: (x, y, z).

by processing the raw radial velocity introduced up to 5 % improvement. The reason why the processed features helped in detecting Pedestrians could be that detecting few, vertically stacked points that are moving helps the model in better detecting such objects as Pedestrians, as such a binary information resembles ground truth.

Though the improvements are modest, they could be valuable in protecting lives, as pedestrians are the most vulnerable road users.

IX. FUTURE WORK

This study pioneers the use of radial velocity information for 3D object detection. This is therefore a preliminary study, which opens a significant breadth of future work.

1) Improving existent results:
   a) Fine-tune the existent use cases: perform hyper-parameter tuning and cross-validation on use case 2 and 3;
   b) Considering other available object types;
   c) Create a confusion matrix to analyse the velocity impact in object classification;
2) Considering new use cases:
   a) Attention mechanisms, where velocity information can be used to select which objects should be considered for further processing and to avoid path collisions (e.g. objects approaching the ego-vehicle);
   b) Potential benefits of the objects radial velocity to vehicle odometry estimation.
3) Real-world validation:
   a) The long-term goal would be to validate the synthetic dataset and the presented results with a real-world dataset acquired with a second-generation LiDAR.

APPENDIX A
DEEP LEARNING MODEL TRAINING DETAILS

Since the acquired synthetic dataset sensor specification and point clouds range resembles KITTI's dataset, PointPillars was trained using the default KITTI configurations available in the OpenPCDet toolbox. The number of epochs was adjusted to 300 for the first use case by the analysis of the loss function, shown in Figure 12. For a matter of time management, the remainder use cases only used 80 epochs. No hyperparameter tuning was performed, which is identified as future work.

REFERENCES

[1] World Health Organization. (2021). *Road Traffic Injuries*. Accessed: Jul. 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries
[2] S. Singh. (2015). *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey*. Accessed: Jul. 2023. [Online]. Available: http://www-nrd.nhtsa.dot.gov/Pubs/812115.pdf
[3] Y. Wu, Y. Wang, S. Zhang, and H. Ogai, "Deep 3D object detection networks using LiDAR data: A review," *IEEE Sensors J.*, vol. 21, no. 2, pp. 1152–1171, Jan. 2021, doi: 10.1109/JSEN.2020.3020626.
[4] D. Bastos, P. P. Monteiro, A. S. R. Oliveira, and M. V. Drummond, "An overview of LiDAR requirements and techniques for autonomous driving," in *Proc. Telecoms Conf. (ConfTELE)*, Leiria, Portugal, Feb. 2021, pp. 1–6, doi: 10.1109/ConfTELE50222.2021.9435580.
[5] D. Bastos, A. Brandão, A. Lorences-Riesgo, P. P. Monteiro, A. S. R. Oliveira, D. Pereira, H. Z. Olyaei, and M. V. Drummond, "Time-frequency range estimation method for pulsed LiDAR," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 1429–1437, Feb. 2023, doi: 10.1109/TVT.2022.3207588.
[6] OpenPCDet Development Team. *OpenPCDet: An Open-Source Toolbox for 3D Object Detection From Point Clouds*. Accessed: Jul. 2023. [Online]. Available: https://github.com/open-mmlab/OpenPCDet
[7] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018, doi: 10.3390/s18103337.
[8] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.
[9] 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.
[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013, doi: 10.1177/0278364913491297.
[11] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2443–2451, doi: 10.1109/CVPR42600.2020.00252.
[12] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11621–11631.
[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3354–3361, doi: 10.1109/CVPR.2012.6248074.
[14] Y. Li and J. Ibanez-Guzman, "LiDAR for autonomous driving: The principles, challenges, and trends for automotive LiDAR and perception systems," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 50–61, Jul. 2020, doi: 10.1109/MSP.2020.2973615.
[15] *Scantinel FMCW LiDAR on CHIP*. Accessed: Jul. 2023. [Online]. Available: https://scantinel.com/technology/
[16] *FMCW LiDAR: The Self-Driving Game-Changer*. Accessed: Jul. 2023. [Online]. Available: https://aurora.tech/blog/fmcw-lidar-the-self-driving-game-changer
[17] *The Future of FMCW LiDAR Technology*. Accessed: Jul. 2023. [Online]. Available: https://www.aeva.com/
[18] *RMCW LiDAR*. Accessed: Jul. 2023. [Online]. Available: https://www.baraja.com/en/blog/rmcw-lidar
[19] X. Peng and J. Shan, "Detection and tracking of pedestrians using Doppler LiDAR," *Remote Sens.*, vol. 13, no. 15, p. 2952, Jul. 2021, doi: 10.3390/rs13152952.
[20] B. Hurl, K. Czarnecki, and S. Waslander, "Precise synthetic image and LiDAR (PreSIL) dataset for autonomous vehicle perception," 2019, *arXiv:1905.00160*, doi: 10.48550/ARXIV.1905.00160.

[21] *Fork GTAV ObjectDetection—Leandro Alexandrino*. Accessed: Jul. 2023. [Online]. Available: https://github.com/leandroalexandrino1995/GTAV_ObjectDetection
[22] *Object Detection Evaluation 2017*. Accessed: Jul. 2023. [Online]. Available: https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d
[23] *Preparing the Input Data Structure*. Accessed: Jul. 2023. [Online]. Available: https://github.com/bostondiditeam/kitti/blob/master/resources/devkit_object/readme.txt
[24] L. Alexandrino, "3D object detection for self-driving vehicles aided by object velocity," M.S. thesis, Departamento de Electrica, Telecomunicas e Infica, Instituto de Telecomunics, Universidade de Aveiro, Aveiro, Portugal, 2022. Accessed: Jul. 2023. [Online]. Available: http://hdl.handle.net/10773/38700
[25] L. Alexandrino, M. Drummond, P. Georgieva, and H. Zahir, "Synthetic automotive LiDAR dataset with radial velocity additional feature—(x,y,z,v)," [Data set], in *Proc. Portuguese Conference on Pattern Recognition (RECPAD)*. Leiria, Portugal: Zenodo, 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7276691

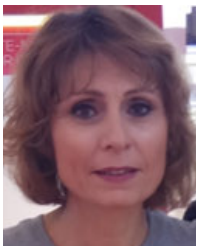
LEANDRO ALEXANDRINO was born in Aveiro, Portugal, in 1995. He received the master's degree in electronics and telecommunications engineering from the University of Aveiro, in 2022. His master's thesis included a curricular internship with Bosch Car Multimedia S.A., Braga, Portugal. Since the conclusion of the master's degree, he joined Bosch Security Systems S.A., Ovar, Portugal, as a Junior Embedded Software Systems Engineer.

HADI Z. OLYAEI received the master's degree in physics from the Azerbaijan University of Tabriz, Iran, and the Ph.D. degree in physics from Instituto Superior Técnico (IST), Portugal, in 2020.

He is currently a Senior Research Engineer with Bosch GmbH, Germany. His research interests include LiDAR sensor technology for autonomous applications, data driven sensor simulation, validation and verification, and LiDAR perception.

ANDRÉ ALBUQUERQUE received the Ph.D. degree in applied physics from the University of Aveiro, in 2017.

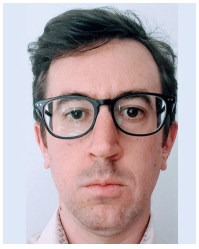
Afterward, he joined Bosch Car Multimedia S.A., Braga, Portugal, as an Optical Engineer in the field of LiDAR technology for automated driving applications. Since 2016, he has been working on the design and simulation of beam shaping and imaging optics for automated driving. His research interests include fiber optics communications, non-linear optics, optical signal processing, optics simulation, automated driving, LiDAR, and vision sensors.



PETIA GEORGIEVA (Senior Member, IEEE) received the master's and Ph.D. degrees in control system engineering from the Technical University of Sofia (TUS) and the Ph.D. degree in computer science from the Faculty of Engineering—University of Porto (FEUP), Portugal, in 2003.

She is currently a Professor in machine learning with the Department of Electronics Telecommunications and Informatics (DETI), University of Aveiro, Portugal, and a Senior Researcher with the Institute of Electronics Engineering and Telematics of Aveiro (IEETA), Portugal. She has held several visiting positions: A Visiting Faculty Member of Carnegie Mellon University (CMU), Silicon Valley, in 2019 (Fall Semester), and CMU-Pittsburgh, in 2012 (Fall Semester); an invited Researcher with Rowan University, Glassboro, NJ, USA, in 2016 (Spring Semester); and an invited Researcher with the School of Computing and Communications, University of Lancaster, U.K., in 2010. Her research is funded by sponsors, such as EU, Portuguese Foundation for Science and Technology (FCT), and industry. Her research interests include machine learning, deep learning, and data mining, with a strong application focus on image processing, autonomous driving, wireless communications, and brain–computer interfaces.

Dr. Georgieva is a Senior Member of International Neural Network Society (INNS).



MIGUEL V. DRUMMOND (Member, IEEE) was born in Porto, Portugal, in 1984. He received the Diploma and Ph.D. degrees in electronics and telecommunications engineering from the University of Aveiro, Aveiro, Portugal, in 2007 and 2011, respectively.

From 2011 to 2014, he was a Research and Development Engineer with Nokia Siemens Networks and then Coriant, where he worked in digital signal processing techniques for high bit rate coherent optical communication systems. Since 2014, he has been an Assistant Researcher with Instituto de Telecomunicações. He has been the Group Leader of the Research Cluster Optical Components and Subsystems, since 2021. He has been involved in several projects funded by the European Commission, Industry, and the European Space Agency, in which he worked on microwave photonics techniques for phased array antennas and on all aspects involving 3D vision for self-driving vehicles.