Deep ConvNet-based Vehicle Detection Using 3D-LIDAR Reflection Intensity Data

Alireza Asvadi, Luis Garrote, Cristiano Premebida, Paulo Peixoto, and Urbano Nunes

Institute of Systems and Robotics, Department of Electrical and Computer Engineering,
University of Coimbra, Coimbra, Portugal,
{asvadi, garrote, cpremebida, peixoto, urbano}@isr.uc.pt

Abstract. This paper addresses the problem of vehicle detection using a less explored LIDAR's modality: the reflection intensity. The reflectivity attribute is related to the type of surface the LIDAR reflection is obtained. A Dense Reflection Map (DRM) is generated from sparse 3D-LIDAR's reflectance intensity, and inputted to a Deep Convolutional Neural Network (ConvNet) object detection framework (YOLOv2 [1]) for the vehicle detection. The proposed approach is the first result using LIDAR's reflection value in the KITTI Benchmark Suite. Although only reflection intensity data is used in the approach presented in this paper, the performance is superior to some of the approaches that use LIDAR's range-value, and hence it demonstrates the usability of LIDAR's reflection for vehicle detection.

Keywords: vehicle detection, 3D-LIDAR reflection, Deep Learning

1 Introduction and Motivation

Vehicle detection is one of the key tasks in intelligent vehicle and intelligent transportation systems technologies. Robust and reliable vehicle detection finds variety of practical applications including collision warning systems, collision avoidance system, autonomous cruise controls, advanced driver assistance systems (ADAS), and in autonomous driving perception systems.

Autonomous vehicles use different types of sensors (*e.g.*, Camera, LIDAR and RADAR) to have a redundant perception system. Color cameras, being the most commonly used sensor, suffer from illumination variations, lacking direct object distance estimation, and inability of vision through the night which restricts the reliability of safe driving. LIDAR and RADAR measure distance by emitting and receiving waves. RADAR, although able to work efficiently in extreme weather conditions, suffers from narrow flied of view and low resolution, which limits its application for object detection and recognition task. In comparison with RADAR, 3D-LIDAR has a precise range measurement and a full 360 degree field of view. Motivated by reduction in their cost and increase in resolution and range, 3D-LIDARs are becoming a reliable solution for scene understanding.

In this paper, we propose a sensory perception solution for vehicle detection (herein called 'RefCN', which is stands for 'Reflectance ConvNet') using a less explored '3D-LIDAR reflection' and a Deep ConvNet-based detection framework (YOLOv2 [1]). The

front view Dense Reflection Map (DRM) is constructed from sparse 3D-LIDAR reflectance data. DRMs run through the trained DRM-based YOLOv2 pipeline to achieve vehicle detection.

The outline of the paper is as follow: Related work is presented in Section 2. The proposed approach is described in Section 3. Experimental results are discussed in Section 4, and Section 5 brings some concluding remarks and future work.

2 Background and Related Work

This section gives a definition and a review of works that relate to LIDAR reflection, and surveys previous work in 3D-LIDAR-based vehicle detection.

2.1 Perception Using 3D-LIDAR Reflection

The LIDAR reflection measures the ratio of the received beam sent to a surface, which depends upon the distance, material, and the angle between surface normal and the ray.

To our knowledge only a few works exist that use LIDAR reflection for the perception tasks. Tatoglu and Pochiraju [2] used LIDAR reflection for point cloud segmentation based on the diffuse and specular reflection behavior. Hernandez et al. [3] detect traffic lanes on the road in urban environments by exploiting the LIDAR reflection of lane marking.

2.2 3D-LIDAR-based Vehicle Detection

This subsection gives a concise overview of vehicle detection approaches using 3D-LIDARs in IV and ITS domains.

Behley et al. [4] performed object detection with a hierarchical segmentation of 3D-LIDAR points followed by bag-of-word (BoW) classifiers. Wang and Posner [5] discretized LIDAR points and reflectance values into a Voxel grid. A 3D sliding window with a linear SVM classifier is used for obtaining detection scores. Li et al. [6] used a 2D Fully-Convolutional Network (FCN) in a 2D point map (top-view projection of 3D-LIDAR range data) and trained it end-to-end to build a vehicle detection system. Li [7] extended it to a 3D Fully-Convolutional Network (FCN) to detect and localize objects as 3D boxes from LIDAR point cloud data. Gonzalez et al. [8] use images and LIDAR-based depth maps with HOG and Local Binary Patterns (LBP) features. They split the training set in different views to take into account different poses of objects and train a Random Forest (RF) of local experts for each view for object detection. Chen et al. [9] use LIDAR range data and images. A top view representation of point cloud is used for 3D proposal generation. 3D proposals were projected to color image, LIDAR's top and front views. A ConvNet-based fusion network is used for 3D object detection. Oh et al. [10] use segmentation-based object proposal generation from LIDAR depth maps and color images. They use decision level fusion to combine detections from two independent ConvNet-based classifiers in the depth map and color image. Table 1 provides a review of vehicle detection approaches using 3D-LIDAR data.

Table 1. Some related work on 3D-LIDAR-based Vehicle Detection.

Reference	Modality	Representation	Detection Technique
Behley et al. [4]	Range	Full 3D: Point cloud	Hierarchical Seg. + BoW
Wang and Posner [5]	Range + Reflec.	Full 3D: Voxel	Sliding-window + SVM
Li et al. [6]	Range	Top view	2D-FCN
Li [7]	Range	Full 3D: Point cloud	3D-FCN
Gonzalez et al. [8]	Image + Range	Front view	Sliding-window + RF
Chen et al. [9]	Image + Range	Front + Top views	Top view 3D Prop. + ConvNet
Oh et al. [10]	Image + Range	Front view	Seg.-based Prop. + ConvNet

While most of the 3D-LIDAR-based vehicle detection systems were built on range data, in contrast, in this paper, a less probed 3D-LIDAR reflection return is used as a medium for the vehicle detection. The front view Dense Reflection Map (DRM) is generated and used with YOLOv2, and is shown that the DRM can be useful for the vehicle detection purpose.

3 RefCN: Vehicle Detection Using 3D-LIDAR Reflection and YOLOv2 Object Detection Framework

The architecture of the proposed RefCN vehicle detection system is shown in Fig. 1. The 3D-LIDAR point cloud is projected to the camera coordinate and a Sparse Reflectance Map (SRM) is generated. The SRM is up-sampled for getting a Dense Reflectance Map (DRM). The DRM inputted to the trained DRM-YOLO to detect vehicles. The RefCN has two steps: (1) DRM construction and (2) DRM-based YOLOv2 object detection framework, described as follow.

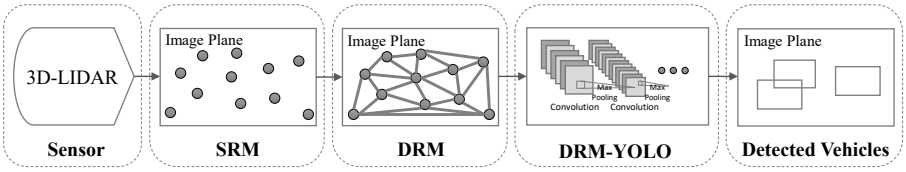

Fig. 1. The pipeline of the RefCN.

3.1 Delaunay Triangulation-based DRM Reconstruction

The DRM is generated by projection of the point cloud on the image plane, triangulation and interpolation as described in the following subsections.

3D-LIDAR Image Projection Point cloud with reflection intensity $P = \{X, Y, Z, I\}$ is filtered to the camera's view field, and projected onto the 2D-image plane using,

$$P^* = \overbrace{P2 \times R0 \times VtC}^{\text{Projection Matrix}} \times P \tag{1}$$

where $P2$ is the projection matrix from the camera coordinate system to the left color image, $R0$ is the rectification matrix, and VtC is LIDAR to camera coordinate system projection matrix. Considering $P^* = \{X^*, Y^*, Z^*, I^*\}$, using the row and column values $\{X^*, Y^*\}$ accompanied with reflectance data I^*, a compact Sparse Reflectance Map (SRM) is computed, which has a lower density than the image resolution.

Delaunay Triangulation (DT) The Delaunay Triangulation (DT) is used for mesh generation from the row and column values $\{X^*, Y^*\}$ of the projected 3D-LIDAR points P^*. The DT produces a set of isolated triangles $\Delta = \{\delta_1, \cdots, \delta_n\}$, each triangle δ composed of three vertices $n_k, \{k : 1, 2, 3\}$, useful for building the interpolating function $f(\cdot)$ to perform interpolation on reflectance values I^*.

Interpolation of Sparse Reflectance Map (SRM) The (missing) value $\mathscr{I}$ of a pixel $\mathscr{P}$ which lie within a triangle δ, is estimated by interpolating reflectance values of the surrounded triangle vertices $n_k, \{k : 1, 2, 3\}$ using Nearest Neighbor interpolation (which means selecting the value of the closest vertex), ending up in a DRM.

$$\mathscr{I} = f(\underset{n_k}{\text{argmin}} ||\mathscr{P} - n_k||), \quad \{k : 1, 2, 3\} \tag{2}$$

For more details on the interpolation of scattered points, please refer to [11].

3.2 DRM-based YOLOv2 Object Detection Framework

You Only Look Once (YOLO) [12] models object detection as a regression problem. The most-recent version of YOLO, denoted as YOLOv2 [1], is used in this paper.

The DRM is divided into 13×13 grid regions, where each grid cell is responsible for predicting five object BB centers with their associated confidence scores. A convolutional network runs once on the DRM to predict object BBs. The network is composed by 19 convolutional layers and 5 max-pooling layers. YOLO looks at the whole DRM during training and test time; therefore, in addition to vehicle appearances, its predictions are informed by contextual information in the DRM. The constructed DRM run through the trained DRM-based YOLOv2 pipeline to achieve vehicle detection. Fig. 2 shows different steps of the RefCN.

4 Experimental Results and Analysis

For the RefCN evaluation, quantitative and qualitative experiments using the KITTI dataset [13] was performed.

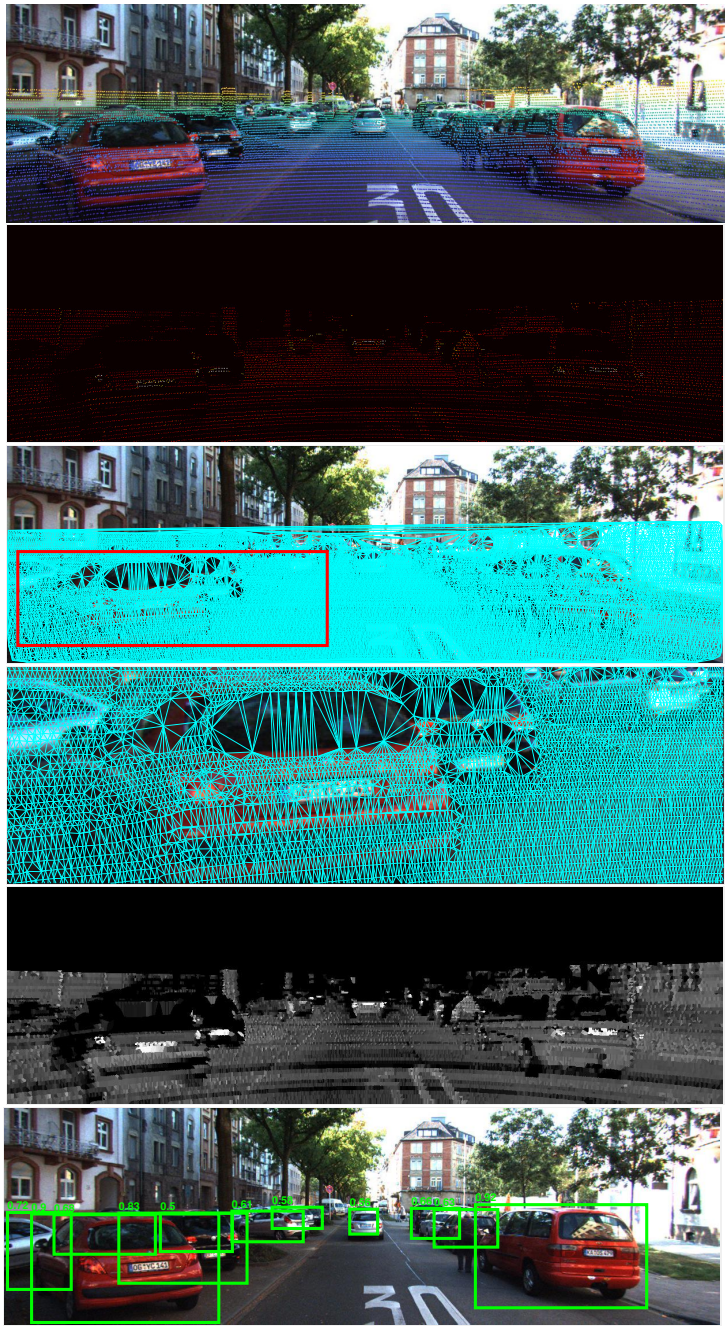

Fig. 2. Illustration of the RefCN process. Top to bottom: a color image with superimposed projected LIDAR points. The SRM with color coded reflectance values. The generated 2D triangulation. The zoomed area within the red box. The fifth image represents the constructed DRM. The last image shows the vehicle detection result with confidence scores.

4.1 KITTI Object Detection Dataset

The KITTI object detection 'training dataset' (containing 7,481 frames) was partitioned into two subsets: 80 % as training set (5,985 frames) and 20 % as validation set (1,496 frames). The 'Car' label was considered for the evaluation.

4.2 Evaluation Metrics

Following KITTI's assessment methodology, the PASCAL VOC intersection-over-union (IOU) metric on three difficulty levels was used as the evaluation criterion with an overlap of 70% for car detection. The overlap rate in 2D is given by,

$$IOU = \frac{area(\text{2D-BB} \cap \text{2D-BB}^g)}{area(\text{2D-BB} \cup \text{2D-BB}^g)} \tag{3}$$

The precision-recall curve and Average Precision (AP), which corresponds to the area under the precision-recall curve, were computed and reported over easy, moderate and hard data categories to measure the detection performance.

4.3 Experimental Setup and Computational Analysis

The experiments were run on a computer with a Hexa-core 3.5 GHz processor, powered with a GTX 1080 GPU and 64 GB RAM under Linux. The MATLAB *scatteredInterpolant* function and the YOLOv2[1] [1] 416×416 detection framework, written in C, were used in the RefCN implementation.

YOLOv2 Training with DRMs For training, as an initial weight, convolutional weights of the pre-trained ConvNet on ImageNet are used. Next, the DRM-YOLOv2 is fine-tuned for 80,200 iterations using stochastic gradient descent with learning rate of 0.001, 64 as batch size, weight decay of 0.0005 and momentum of 0.9 by training on KITTI DRMs to adapt it to the DRM-based vehicle detection.

Computational Complexity The implementation details and the computational load of DRM generation and YOLO detection steps are reported in Table 2. The DRM generation is the most time-consuming part which can be further reduced by reimplementation in C/C++.

Table 2. The processing time (in seconds) of the RefCN.

Impl. Details	Proc. Time	Environment
DRM Generation	1.403	MATLAB
YOLO Detection	0.015	C

[1] https://pjreddie.com/darknet/yolo/

4.4 Quantitative Results

Two sets of quantitative experiments were conducted to assess the performance of the RefCN.

Sparse Reflectance Map (SRM) vs DRM The RefCN is trained on training set and evaluated on the validation set. As it can be seen in Fig. 3 and Table 3, the results show that the DRM considerably improves the detection performance.

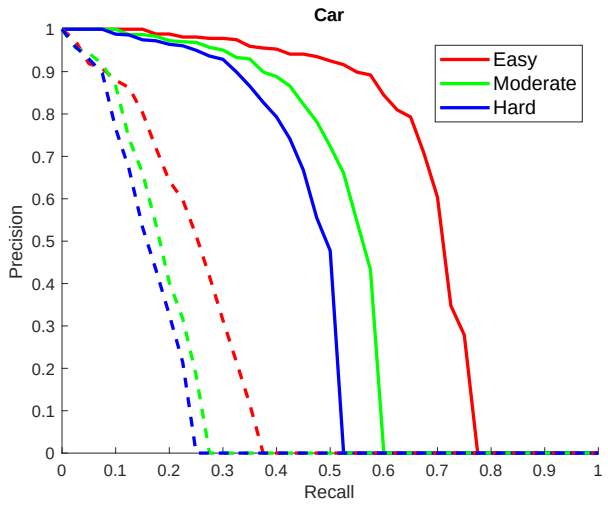

Fig. 3. Precision-Recall on KITTI Validation-set (Car Class) using SRM (dashed lines) and DRM (solid lines).

Table 3. Detection Accuracy with SRM vs DRM on Validation-set.

Input Data	Easy	Moderate	Hard
SRM	23.45 %	17.57 %	15.57 %
DRM	67.69 %	51.91 %	44.98 %

Comparison with the State-of-the-art To compare with the state-of-the-art, the RefCN is trained on the full KITTI object detection 'training dataset' and evaluated on the test set. Results are reported in Fig. 4 and Table 4. Although the RefCN uses only reflection-value, it surpasses some of the approaches that use LIDAR's range-value, and hence it demonstrates the usability of LIDAR's reflection for the vehicle detection.

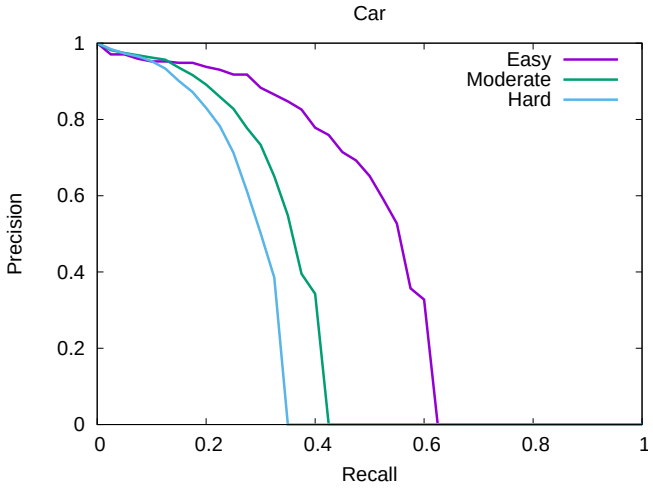

Fig. 4. Precision-Recall on KITTI Test-set (Car Class).

Table 4. RefCN Evaluation on KITTI Test-set.

Approach	Moderate	Easy	Hard
MV3D (LIDAR) [9]	79.17 %	89.01 %	78.09 %
3D FCN [7]	75.83 %	85.54 %	68.30 %
VeloFCN [6]	53.45 %	70.68 %	46.90 %
Vote3D [5]	48.05 %	56.66 %	42.64 %
RefCN (proposed)	35.72 %	50.28 %	29.86 %
CSoR [14]	26.13 %	35.24 %	22.69 %
mBoW [4]	23.76 %	37.63 %	18.44 %

4.5 Qualitative Results

Figure 5 shows some of the representative qualitative results with many cars in the scene. A video result is available at https://goo.gl/ovQYeq. As can be seen, for most cases, the RefCN correctly detects target vehicles.

5 Conclusions, Key Findings and Future Work

To fulfill the requirements of the redundant perception system, we presented a methodology for using LIDAR reflection return values for vehicle detection, and we proved the benefits of the system through quantitative and qualitative experiments. As a direction for the future research, based on our observations, as shown in Fig. 6, traffic signs and license plates have high reflectance values. The DRM can be explored for car license plate and traffic sign detection.

Fig. 5. Example screenshots of RefCN results. Detection results are shown, as green BBs in the color-images (top) and DRMs (bottom) compared to the ground-truth (dashed-magenta). Notice that the depicted color-images are shown only for visualization purpose.

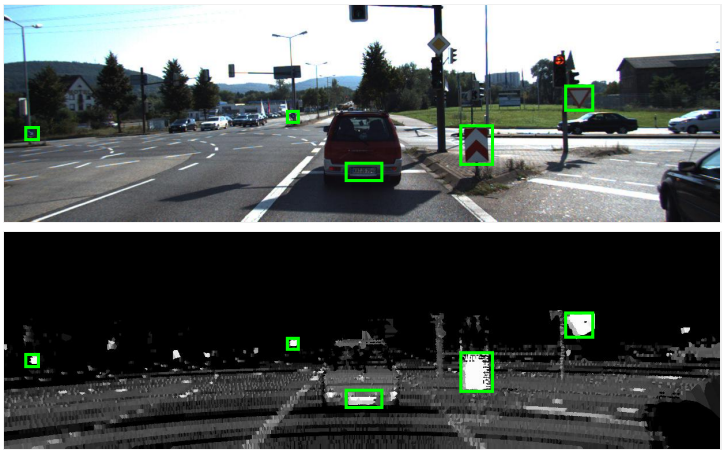

Fig. 6. The effect of retroreflectivity of car license plates and traffic signs in the DRM and the corresponding color-image. The green BBs show a car license plate and traffic signs.

Acknowledgments

This work has been supported by “AUTOCITS - Regulation Study for Interoperability in the Adoption of Autonomous Driving in European Urban Nodes” - Action number 2015-EU-TM-0243-S, co-financed by the European Union (INEA-CEF); and FEDER through COMPETE 2020 program under grants UID/EEA/00048.

References

1. Redmon, Joseph, and Ali Farhadi. ”YOLO9000: better, faster, stronger.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.
2. Tatoglu, Akin, and Kishore Pochiraju. ”Point cloud segmentation with LIDAR reflection intensity behavior.” Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012.
3. Hernandez, Danilo Caceres, Van-Dung Hoang, and Kang-Hyun Jo. ”Lane surface identification based on reflectance using laser range finder.” System Integration (SII), 2014 IEEE/SICE International Symposium on. IEEE, 2014.
4. Behley, Jens, Volker Steinhage, and Armin B. Cremers. ”Laser-based segment classification using a mixture of bag-of-words.” Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013.
5. Wang, Dominic Zeng, and Ingmar Posner. ”Voting for Voting in Online Point Cloud Object Detection.” Robotics: Science and Systems. 2015.
6. Li, Bo, Tianlei Zhang, and Tian Xia. ”Vehicle detection from 3d lidar using fully convolutional network.” Robotics: Science and Systems. 2016.
7. Li, Bo. ”3D Fully Convolutional Network for Vehicle Detection in Point Cloud.” Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on. IEEE, 2017.
8. Gonzlez, Alejandro, et al. ”On-board object detection: Multicue, multimodal, and multiview random forest of local experts.” IEEE transactions on cybernetics (2016).

9. Chen, Xiaozhi, et al. "Multi-view 3d object detection network for autonomous driving." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.
10. Oh, Sang-Il, and Hang-Bong Kang. "Object detection and classification by decision-level fusion for intelligent vehicle systems." Sensors 17.1 (2017): 207.
11. Amidror, Isaac. "Scattered data interpolation methods for electronic imaging systems: a survey." Journal of electronic imaging 11.2 (2002): 157-176.
12. Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
13. Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.
14. Plotkin, Leonard. Pydriver: Entwicklung eines frameworks fr rumliche detektion und klassifikation von objekten in fahrzeugumgebung. Diss. Bachelors Thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2015.