Impact of miscalibration and camera noise in an active stereo-based LiDAR in 3D object detection

Roberto O. Graça[12]
robertograca@ua.pt
Miguel V. Drummond[12]
mdv@av.it.pt
Paulo M. N. P. Monteiro[12]
paulo.monteiro@ua.pt

[1] Department of Electronics, Telecommunications and Informatics
University of Aveiro
Aveiro, Portugal

[2] Instituto de Telecomunicações
University of Aveiro
Aveiro, Portugal

Abstract

LiDARs are a powerful technology capable of capturing high-resolution 3D point clouds of the surroundings. However, because it is expensive and immature, LiDAR is not yet a viable option for car manufacturers for mass-production.

We propose an alternative LiDAR sensor made of affordable and mass-produced components, which works like an active stereo system, however with physically detached cameras.

This paper focuses on the fact that such a setup is prone to being miscalibrated and is also affected by camera noise. The objective of this paper is to observe whether camera miscalibration and noise are harmful to 3D object detection.

1 Introduction

According to studies from the WHO [7], 1.3 million people die each year as a result of road traffic crashes. Factors such as speeding, distracted driving and driving under the influence of alcohol or other psychoactive drugs are some of the major risk elements that cause these numbers to be so high. To mitigate these risks, one possible solution is offsetting the human functions, such as, driving and parking to the vehicle.

Of all the groups of electronics that can assist in this realm, the vision sensor group is one of the most important. Vision sensors enable the vehicle to perceive the environment and make decisions accordingly. In this group there are: cameras, LiDARs and RaDARs. From these the LiDAR is the most capable of acquiring a 3D map of it's surroundings, enabling the autonomous vehicle to have 3D vision. However, due to it's expensiveness and immaturity it is still not a viable option for car manufacturers.

In this work, we propose a LiDAR based on an active stereo setup made with affordable, mass-produced components.

1.1 Proposed Stereoscopic LiDAR Setup

The setup makes use of active stereo, by using two cameras with a distance of 1.2 meters between each other and a dot pattern projector that generates a dot pattern similar to a Velodyne HDL-64E [4]: 64 lines of dots with two consecutive dots of the same line diverging by 0.08 degrees. And thus, enabling the setup to make a point cloud of the surroundings. The setup is in a simulated environment and uses a synthetic dataset [1] [2].

Some factors are needed to be taken in consideration: noise, calibration, correspondence and centroiding. This paper focuses on noise and calibration. Physically unattached cameras are prone to become miscalibrated over time. Camera noise will vary depending on the camera sensor used. Because of that, there is a need to know if camera miscalibration and noise impair the performance of 3D object detection.

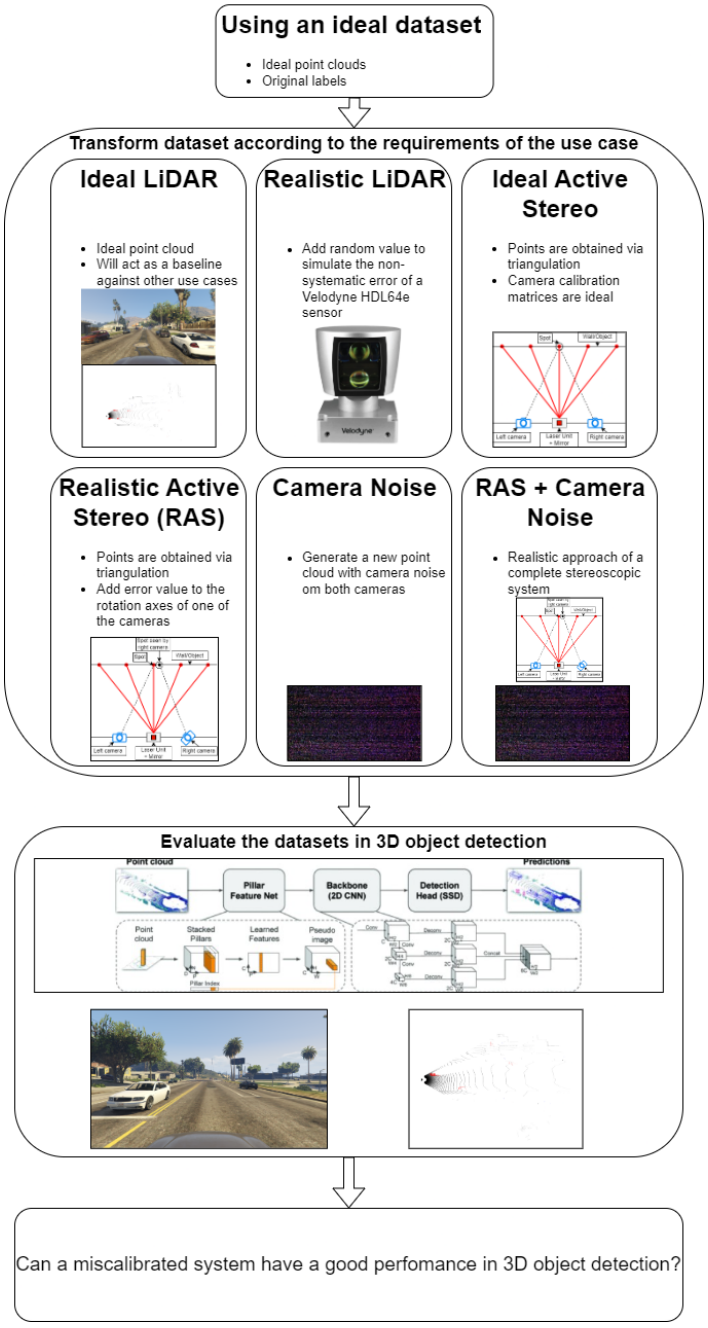

Figure 1: Methodology and main objective in this work.

2 Methodology

The methodology behind this work is detailed in Figure 1. The use cases are as such:

1. The first use case is the ideal scenario and the dataset for the model is used as is. This model acts as a baseline.
2. The second use case depicts a realistic LiDAR sensor. The dataset suffers a non-systematic change, which means that every point in a point cloud will be randomly shifted between 0 and 10 centimeters from their original position.
3. The third use case is the first usage of the proposed setup. A stereoscopic LiDAR means that every point is calculated via triangulation, but in this case, since both intrinsic and extrinsic camera matrices are ideal, the points are only shifted by the amount of the residual triangulation error, which is in the order of the picometers. The results are expected to be identical to the first use case.
4. In the fourth use case we added error to the rotation axes of the extrinsic camera matrices. The selected error value is the Field-of-View per Pixel value, which is approximately 0.24 degrees. This error is only applied to the right camera, while the left camera is left with an ideal

rotation matrix.

5. In the fifth use case we introduced camera noise. To reduce computation complexity, we first created a table of values to add during the processing of the point cloud. The noise value is calculated using a model from [3]. The model will depend on five free parameters that vary from camera-to-camera. They are: the total quantum efficiency; the read noise variance (or its standard deviation); the dark current; the sensitivity; and the bit-depth of the camera. The model starts by converting the electric field to units of photons. Then, it computes the photon shot noise and the number of photo-electrons, the number of photo-electrons is the product of the quantum efficiency with the shot noise. After that, it needs to simulate the read noise and dark current that are the main sources of dark noise. The next step is to convert each pixel from electrons to ADU. To do this, the number of electrons after the addition of read noise is multiplied by the sensitivity and it's maximum upper value is $2^k - 1$, where k is the camera's bit-depth. The last step is to add a baseline to prevent the number of ADU's from becoming negative at low input signal.
6. The sixth use case is a combination of the techniques described in the fourth and fifth use cases.

PointPillars [5] was the 3D object detection Deep Learning model chosen due to it's robustness and fast training time while having good performance. And also because of it's availability in the OpenPCDet toolbox [6], that simplifies the model configuration and optimization.

The metrics employed by the KITTI benchmark are widely adopted for perception tasks. Because of that, the models are evaluated in 3D object detection using the Average Precision metric, with 40-point interpolation, for three KITTI difficulty levels: Easy, Moderate and Hard.

Cross-validation with 4 folds was used in the first and sixth use cases. The remaining use case were validated using only the first fold. If the results are consistent in both use cases, then the rest of the use cases are also consistent, therefore, due to our time constraints this was the best option to perform evaluation of the models.

3 Results

The evaluation of the models was done using both training and testing frames to validate the models. Besides that, the labels across all the use cases were the original labels.

As we can observe from Table 1, when any model is evaluated on the dataset generated for the same use case, the performance holds high. However when confronting the baseline scenario, some score drops occur, mainly in the fourth and sixth use cases. When changes occur, starting from the second use case, we can determine that, although there was a small change in the dataset, the model is robust enough to handle it. However in the fourth and sixth use cases, the changes are more noticeable. The score drops up to almost fifty points. When error is added to the rotation axes of the extrinsic camera matrices, some sections of the point cloud expand, while others shrink, this leads to the object dimensions not being constant and, therefore, not being in a box shape with certain dimensions. This could mean that objects will not be able to be classified correctly by the model. Another explanation might be, because the models, from the fourth and sixth use cases, were trained with this error, they are able to compensate the displacement caused by the error, that is, the models 'see' where an object is after the displacement, but know where it should be if the displacement did not happen, which is not the case for the first use case model. Also in the sixth use case, because of the introduction of camera noise, the score increases by a small amount. One possible explanation can be that when the point cloud becomes fuzzy the model becomes less rigid and leaves more room for the detection of objects.

4 Conclusion

In this paper, we analyzed how different impairments affect the precision of an automotive LiDAR based on active stereo.

From the results, we can observe that 3D object detection operates well provided that the model is trained in the same conditions. While the other way around shows that the introduction of a non-constant error is difficult to handle for models that are not expecting it. We can conclude that a non-ideal calibrated model can have a good performance in 3D object detection, but only if the calibrated error is taken into consideration when training the models.

Improving the overall results can be done by increasing the robustness of the model. This can be done in two ways: increasing the resilience of the 3D object detection or keeping the miscalibration within limits. The former one can be accomplished by adding frames that range several types of errors: systematic and non-systematic. For example, obtaining frames using the proposed setup with other calibration errors, agglomerating them and using them to train the model. The latter one can be accomplished by resorting to surrounding features for estimating the camera parameters and correcting them if needed.

UC for which the model was tested \ UC for which the model was trained	UC1	UC2	UC3	UC4	UC5	UC6
UC1	95.4 89.9 87.3					
UC2	90.8 83.0 79.9	95.1 89.3 84.6				
UC3			95.3 89.6 87.0			
UC4	59.0 35.3 33.4			95.2 88.4 84.0		
UC5	94.8 78.5 74.1				95.0 89.0 84.3	
UC6	68.0 43.4 40.5					95.1 86.2 81.5

Table 1: Results for all difficulties. Every cell, from top to bottom, presents the average precision (%) for Easy, Medium and Hard difficulties.

References

[1] Leandro Alexandrino. 3d object detection for self-driving vehicles aided by object velocity. Master's thesis, University of Aveiro, 2022. URL http://hdl.handle.net/10773/38700.

[2] Leandro Alexandrino, Miguel Drummond, Petia Georgieva, and Hadi Zahir. Synthetic automotive LiDAR dataset with radial velocity additional feature - (x,y,z,v), October 2022. URL https://doi.org/10.5281/zenodo.7276691.

[3] Kyle Douglass. Modeling noise for image simulations. https://kmdouglass.github.io/posts/modeling-noise-for-image-simulations/, 2017. Accessed: 2023-08-08.

[4] Hypertech. High definitian lidar hdl-64e. https://hypertech.co.il/wp-content/uploads/2015/12/HDL-64E-Data-Sheet.pdf. Accessed: 2023-08-30.

[5] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019. URL https://doi.org/10.48550/arXiv.1812.05784.

[6] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. Accessed: 2023-08-30.

[7] WHO. Road traffic injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, 06 2022. Accessed: 2023-07-11.