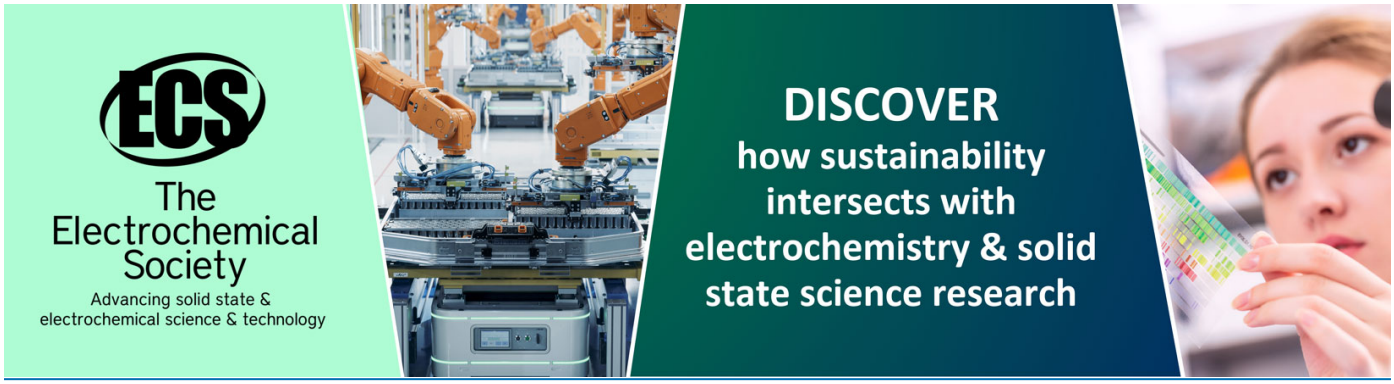
Spherical representation of light detection and ranging data for three-dimensional object detection



Spherical representation of light detection and ranging data for three-dimensional object detection

J Pamplona[1], J Herrera-Ramirez[1], and C Madrigal[2]

[1] Laboratorio de visión Artificial y Fotónica, Instituto Tecnológico Metropolitano, Medellín, Colombia

[2] Grupodot, Medellín, Colombia

E-mail: josepamplona212620@correo.itm.edu.co

Abstract. Autonomous vehicles are one of the most attractive applications for light detection and ranging sensors, where they help with scene understanding. For this understanding, object detection is crucial, and it must be done in a frame by frame basis. This detection on a single frame is a challenging task due to the sparse and disordered nature of the information. This paper presents an alternative spherical representation for this data aiming to improve object detection. This proposal registers the light detection and ranging data in a 2-dimensions angle map using most of the 3-dimensions points in three layers, adding reflectivity information, and a logarithmic representation of distance. For evaluating this representation, we employed an object detector based on the algorithm: you only look once version 3, and we used a public reference dataset of 3-dimensional objects for training. This framework yielded a classification accuracy of 85.9% and 74.5% of intersection over union factor when estimating seven classes simultaneously. This approach presents an alternative for processing this data that helps to benefit the most from the light detection and ranging information with high accuracy, helping in the reduction of the associated risks of autonomous vehicles.

1. Introduction

Since the high definition (HD) light detection and ranging (LiDAR) became popular, a great number of applications have been using it. Object detection in three dimensions (3D), is one of these applications and it is getting a lot of attention in the areas of city model generation, power lines monitoring [1] and autonomous driving [2]. Autonomous vehicles are receiving special attention these days. Car crash is the first cause of death for people between 5 and 29 years [3]. To avoid the collisions, an autonomous vehicle should get an scene understanding and the LiDAR sensor is one of the main devices in this task. The LiDAR information have to be used in real-time, but a single frame of LiDAR data has limited information about the environment. This drawback added to a variable point density and occlusions [2], makes the object detection a challenging task in context of autonomous driving.

In general, object detection is a well known task using red, green, blue (RGB) images. In the last 8 years, the deep learning techniques have demonstrated superiority over the classic machine learning methods [4]. In the case of the object detection applications over 3D LiDAR information, it does not offer the same performance as the detection using RGB images, however, as a topic in development it still has a wide range and of improvement.

Like object detectors in RGB images, The initial solutions proposed for 3D object detectors with handcrafting features were surpassed by the deep learning methods rapidly. Even when there are some deep learning methods like PointNet [5], that use directly LiDAR data. Most of 3D object detection methods uses a transformation of the LiDAR data in order to sort the information getting better results. A good example is VoxelNet [2]. This method uses a 3D representation sorted by voxels and processed in a 3D convolutional architecture. This representation choose some points into each voxel, and doing so, it discards a lot of information.

Another methods stands by using LiDAR data transformed on 2D representations. In the SqueezeSeg method is presented an architecture for semantic segmentation. In this method is used a representation with a spherical coordinates mapping, cartesian coordinates and the reflectivity information to configure a 2D with five channel feature tensor [6].

To take advantage of the gained experience by deep learning methods on RGB images, some methods have explored a transformation named bird eye view (BEV) which is used with 2D object detectors pre-trained with original RGB images like does Beltran, *et al.* in [7]. BEV projects all the points on a (x, y) grid and code the information of the points into each grid box in 3 channel pixel, alike RGB pixels [7]. In this representation the small objects get represented in few pixels. This prevents the objects to be detected as can be evidenced in [8].

In this paper we propose a spherical representation of LiDAR data presented in a RGB format to take advantage of all the information in the LiDAR data and the gained experience by the researchers on object detection when using RGB images. In this way, this alternative allows a better understanding of the scene providing more accuracy on the detection of classes of importance in the autonomous vehicles application as well as in any other application using LiDAR data. We evaluate this proposal using You Look Only Once version 3 (YOLOv3) architecture as object detector over a public reference data-set for 3D object detection.

The outline of the paper is as follows. Section 2 explain the spherical representation. The detection method and dataset conditioning is described in section 3. The section 4 presents the experiments and results. Finally, we conclude on section 5.

2. Spherical representation of light detection and ranging data

The high definition LiDAR has 64 semiconductor lasers in a vertical arrangement. These rotate around the vertical axis of the device as described in Figure 1(a). Each laser takes over 2000 measurements per rotation [9]. As shown in Figure 1(b), each laser measurement can be associated with the angle between the sensor and the Z axis (ϕ). The device angle around Z axis is associated with (θ), and the distance measured by the LiDAR with (ρ), as it was presented in [6]. These values establish the spherical coordinates for each measurement, as shown as an example by the red dot in Figure 1(b).

A row is defined by approximately 2032 distance values (ρ) measured by each laser in a complete rotation. The 64 vectors define a matrix with shape $64 * 2032$ configuring a spherical frame of LiDAR data. Since the data given by the LiDAR is presented in a 3 axis cartesian format, each point should be transformed using Equation (1), Equation (2), and Equation (3). Where x, y *and* Z are the 3D cartesian coordinates.

$$\rho = \sqrt{x^2 + y^2 + z^2}, \tag{1}$$

$$\theta = \arctan(y/x), \tag{2}$$

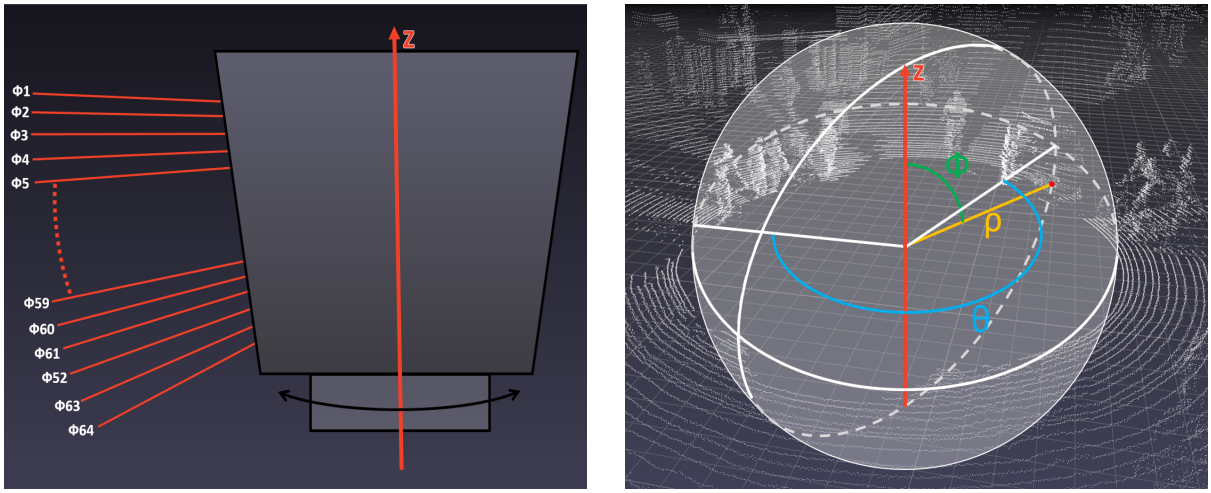
$$\phi = \arccos(z/\rho). \tag{3}$$

The high definition LiDAR also provides a reflectivity value for each measurement which can be added as a second channel to the representation. To take advantage of the good results obtained by deep learning object detection methods in RGB images, a third channel is proposed.

It is composed using a logarithmic representation of the point distances as shown in Equation (4). Where ρ is the distance measured for the specific point, and ρ_{max} is the maximum distance of the corresponding frame. This channel aims to a better discrimination on distant objects where the distance gap is increased.

$$\rho_{logarithm} = \log(\rho_{max} - \rho), \tag{4}$$

All the 3 Channels were normalized to be stored as an 8 bit RGB image. The Figure 2 is a resulting image of the 360 degrees spherical representation of a single frame of LiDAR data. In Figure 2 can be identified the reflectivity in the red channel and the farthest points highlighted by the green channel.



(a) (b)

Figure 1. Spherical representation of a single LiDAR data point. (a) LiDAR HD laser layout, and (b) ϕ, θ *and* ρ in a spherical representation.

Figure 2. Image of the spherical representation of one LiDAR data frame.

This RGB presentation of the spherical representation have two cons. First the RGB image uses the column which represents the angle $\theta = 0$ as the first column. This cut the front of the vehicle scene in two. This situation divide a lot of labeled objects due all these objects are in front of the vehicle scene. The second disadvantage of the Figure 2 is the ratio between rows and cols. These were resolved by rotating the entire point cloud 180^o arround the Z axis and limiting the representation to the zone where the labeled objects are (between $70^o and -70^o$). The Figure 3 is the result of the described modification.

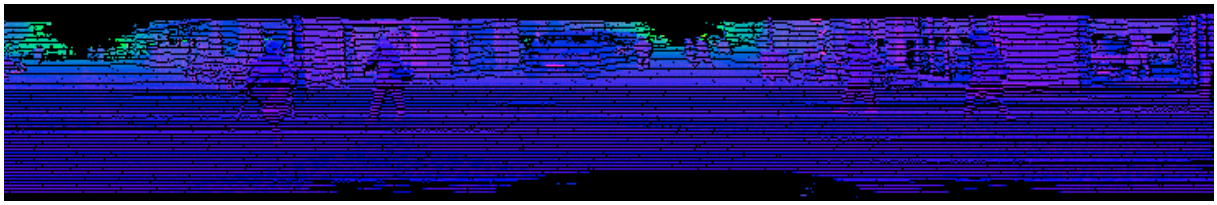


Figure 3. Image of the spherical representation after the rotation and crop process.

3. Training details

3.1. Network Details

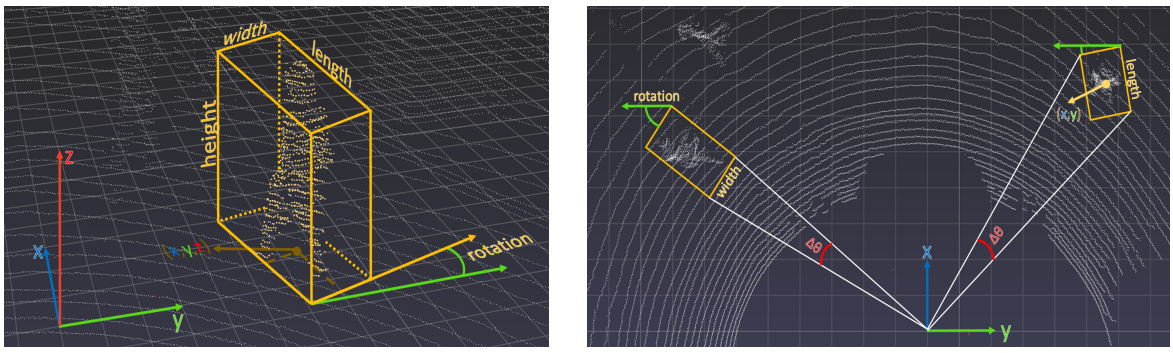
You only look once (YOLO) is one of the most popular deep learning object detection algorithms. The method is composed by a convolutional neural network (CNN) which simultaneously determines the bounding boxes and the classes of the detected objects. The main contribution of YOLO is the region proposal network (RPN) which predicts the bounding boxes by simple regression. This network computes some offsets to build the detected object bounding boxes using box sizes predefined for the specific detection application. The RPN is included in the same CNN architecture getting a very low latency [10].

There are a lot of variations of this architecture which can be useful in a wide range of applications. In the last actualization of the method, YOLOv3 [11], some configurations can make it faster or more precise. To asses the spherical representation, two configurations are used: the Tini-YOLO configuration, which has 7 convolutional layers and the region proposal layers; and a complete version of YOLOv3, which has 53 convolutional layers using some residual layers and 3 different region proposal layers.

3.2. Dataset details

Our experimental setup is based on the 3D object detection benchmark data-set from the KITTI data-set (a project of Karlsruhe Institute of Technology and Toyota Technological Institute) [12]. The spherical representation was used in 7481 LiDAR frames. This data-set has over 80.000 labeled objects, all of them in the front of the vehicle. The spherical representations are cropped to fit the region where the labeled objects are.

The original object labels are presented as bounding boxes in cartesian format as presented in Figure 4(a), where three blocks of information can be observed: position (x, y, z), dimensions (height, width and length) and rotation (angle between π and $-\pi$). A function was developed to transform these blocks into center (ϕ, θ) and dimensions $(\Delta\phi, \Delta\theta)$. The spherical center is computed using the cartesian center, adding to the z magnitude of this point the half of the object height and applying the Equation (2) and Equation (3) to the resulting point. $\Delta\phi$ is computed with the cartesian center point, the same point adding height to the z magnitude and computing the difference between the ϕ associated to both points. The $\Delta\theta$ calculation requires a different approach due to its dependence to the object position and rotation as shown in Figure 4(b). The function to compute $\Delta\theta$ uses the θ calculated on the spherical center and the rotation value of the bounding box. With this value conditions the selection of two vertices of the bounding box which determines the spherical width.



(a) (b)

Figure 4. Bounding boxes representation.

3.2.1. Data augmentation. We developed a data augmentation procedure to get a better generalization and to take advantage of the Spherical representation. Due to this representation, if some value is added to the θ variable for each data point, a new scene is generated. If the θ value is inverted ($\theta = 360^o - \theta$), a new scene is also generated. Considering this, 10 new scenes were generated from each frame, using five different angles to add and inverting each one of them. At the end of this process, we obtained a data-set with 59000 training examples and 15000 testing examples.

4. Experiments

This augmented data-set with spherical representation, is first tested with the simpler configuration of YOLOv3, the Tiny-YOLO configuration. Most of the objects in this data-set are small objects (less than 10% of the image size), and Tiny-YOLO cannot detect small objects. Because of this reason, in the first training session, after 200 ephocs the mean average precision (mAP) reaches only 38% for classification score on detected objects and 24.4% for intersection over union (IoU) [13] for bounding box prediction score.

The standard configuration of YOLOv3 and its three region proposal layers in different scales gets better results on small objects than Tiny-YOLO. After 200 ephocs, the mAP obtained by this architecture on the spherical dataset is 74.8%. The standard YOLOv3 is capable of detecting objects with sizes between 5 and 90% of the image size. Our dataset has objects under 5%. To detect this small objects, we change the scale of the region proposal layers by doubling the up-sampling. After this modification, the mAP reaches the 85.9% in 200 ephocs. Additional to the mAP, the Recall reaches 86% and the F1-score gets 89%. The predicted bounding boxes gets an average intersection over union of 74.5%. The specific classes results are specified in the Table 1 and a detection example is presented in Figure 5.

Table 1. Average precision for detection on test dataset.

Classes	Car	Van	Truck	Pedestrian	Cyclist	Tram	Miscellaneous
Average precision	89.7%	89.4%	90.3%	77%	86%	82.3%	86.6%

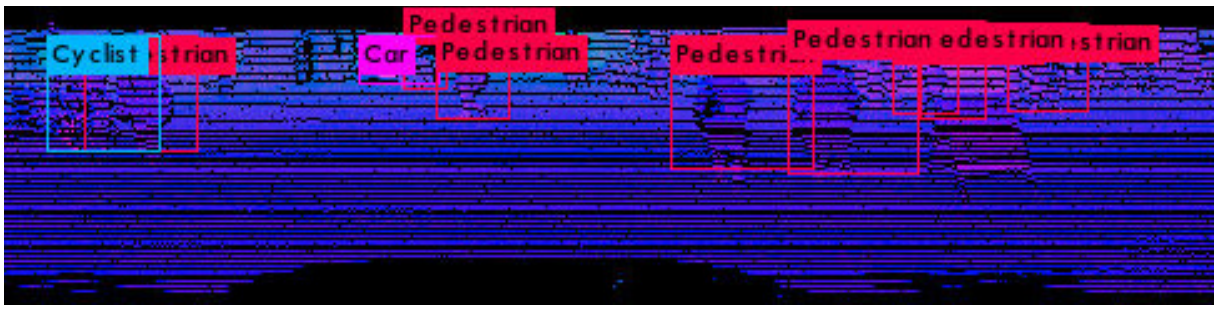


Figure 5. Detected objects after training process.

5. Conclusions

We present a spherical representation of LiDAR data which codes the entire point cloud into an RGB image. This representation was tested for object detection over the KITTI 3D object detection dataset using two different versions of the YOLO architecture. 85.9% in the mAP shows the efficiency of this representation in comparison with the results of methods presented in the 3D object detection benchmark of KITTI. Across all the seven classes we obtained the lowest performance in the pedestrian class, which is the smallest object in the dataset. That is an

expected result due to the small amount of information that represent a pedestrian. According to these results, this representation could be a powerful tool for autonomous vehicle applications. This representation could be used also, on every application with 3D information acquired by any kind of LiDAR sensor. Researches could improve their 3D point clouds analysis using this spherical representation on trained deep learning architectures for original RGB data analysis.

In future work, we will process the information within the bounding boxes to get more specific information about the position and size of the detected objects. This information can be useful for multiple object tracking methods, autonomous path planning and obstacle avoidance methods in autonomous driving environments.

References

[1] Niemeyer J, Rottensteiner F, and Soergel U 2014 Contextual classification of lidar data and building object detection in urban areas *Journal of Photogrammetry and Remote Sensing* **87** 152
[2] Zhou Y and Tuzel O 2018 Voxelnet: End-to-end learning for point cloud based 3d object detection *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City: IEEE) pp 4490-4499
[3] World Health Organization 2018 *Global Status Report on Road Safety* (Geneva: World Health Organization)
[4] Voulodimos A, Doulamis N, Doulamis A and Protopapadakis E 2018 Deep learning for computer vision: A brief review *Computational Intelligence and Neuroscience* **2018(7068349)** 1
[5] Qi C, Yi L, Su H, and Guibas L 2017 Pointnet++: Deep hierarchical feature learning on point sets in a metric space *Advances in Neural Information Processing Systems* **87** 5099
[6] Wu B, Wan A, Yue X and Keutzer K 2018 Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3D lidar point cloud *IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane: IEEE) pp 1887-1893
[7] Beltrán J, Guindel C, Moreno F, Cruzado D, Garcia F and De La Escalera A 2018 Birdnet: A 3D object detection framework from lidar information *21st International Conference on Intelligent Transportation Systems (ITSC)* (Maui: IEEE) pp 3517-3523
[8] Simon M, Milz S, Amende K and Gross H 2018 Complex-YOLO: An Euler-region-proposal for Real-Time 3D object detection on point clouds *European Conference on Computer Vision, Computer Vision – ECCV 2018 Workshops* eds Leal-Taixé L, Roth S (Munich: Springer, Cham) pp 197-209
[9] Schwarz B 2010 LIDAR: Mapping the world in 3D *Nature Photonics* **4** 429
[10] Redmon J, Divvala S, Girshick R and Farhadi A 2016 You only look once: Unified, real-time object detection *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas: IEEE) pp 779-788
[11] Redmon J and Farhadi A 2019 Yolov3: An incremental improvement *ArXiv* **abs/1804.02767** 1
[12] Geiger A, Lenz P, Stiller C and Urtasun R 2013 Vision meets robotics: The KITTI dataset *The Int. Journal of Robotics Research* **32** 1231
[13] Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, and Savarese S 2019 Generalized intersection over union: A metric and a loss for bounding box regression *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach: IEEE) pp 658-666