https://doi.org/10.46335/IJIES.2023.8.3.7 **e-ISSN: 2456-3463**
Vol. 8, No. 3, 2023, PP. 39-42

International Journal of Innovations in Engineering and Science, www.ijies.net

Experimental Analysis of Lidar Point Cloud Data

Tejas Narkhede[1], Sushant Chorghe[2], Punam Kadlag[3]

[1,2,3]PG students, Hochschule Ravensburg-Weingarten University of Applied Sciences, ***Germany***

tejasnarkhede77@gmail.com, sushantchoraghe123@gmail.com, pbkadlag2025@gmail.com

Received on*: 29 March,2023* ***Revised on****: 17 April,2023* ***Published on****: 19 April,2023*

Abstract –*Our goal in this paper is to determine how many points of the Lidar point cloud do come from measurements of the Car. The Data is recorded by different sensors like Camera, Blickfeld Cube sensor and Velodyne Puck sensor for forward and back-word movement of the car[1]. For this case, 240 distinct frames of Blickfeld and Velodyne sensor recorded over 25m distance. We need to compare the number of points we can expect in what distance for all recordings. A typical LiDAR data product is an extensive collection of points cloud data of accurate 3D points with other attributes like intensity and GPS time. There are plenty of tools and softwares to work with LiDAR point cloud data like Desktop-based(QGIS), Web-based(Plas.io).*

Keywords-*Blickfeld Cube sensor, Velodyne Puck sensor, Desktop-based(QGIS), Web-based(Plas.io), intensity, GPS time.*

I -INTRODUCTION

First Use the Provided dataset with 3D bounding box for each sensor and check the specified center and corners points of boundary box. This will give a clear idea of the 3D bounding box. To identify how many points of the lidar point cloud do actually come from measurements of the car, find inverse of the rotation matrix using “numpy. linalg.inv(x)” function and pass this inverse matrix to rotate the point cloud of blickfeld sensor and the corners of Boundary box into the opposite direction than by the yaw value from the ground truth.[3] Rotate the data such that the edges of the bounding box is aligned with the axis. For getting Subpoints of blickfeld Sensor apply for loop on every frames using ”blick.shape[0]” function. Calculate the minimal and maximal value of the corners for the x, y, and z coordinates using “numpy.amax()” function.. The ”dist” function measures the separation between the center of the 3D boundingbox and the coordinate axis origin. Filter all the points generated by sensor which are smaller than the maximum value and larger than the minimum value of corners points in every dimension and Store the points in an array whose values lie between the minimum and maximum values of the corner points. This gives us the count of actual measurements of the car in each recording, and the car inside the 3D bounding box drawn without the outer points. The number of points inside the 3D bounding box and the distance for all 240 different frames is stored in another array.[2] Use the plot function to draw the number of points inside the 3D bounding box and the distance of the car for the

https://doi.org/10.46335/IJIES.2023.8.3.7 e-ISSN: 2456-3463
Vol. 8, No. 3, 2023, PP. 39-42

Blickfeld Cube sensor. Plot the graph using the same process for the Velodyne Puck sensor.

II. METHODOLOGY

A) Using the Blickfeld Cube Sensor

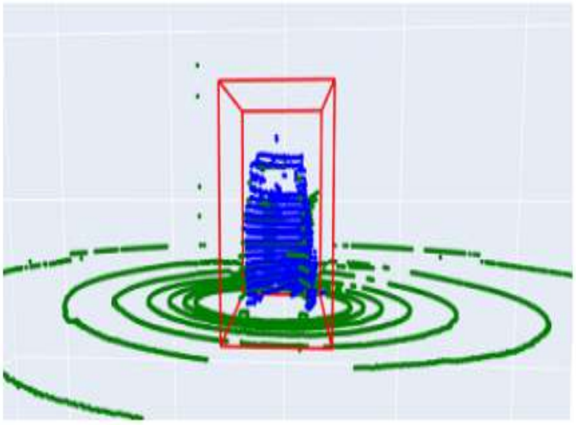

Fig.1: Blickfeld Cube Sensor 3D view for 162 FrameID

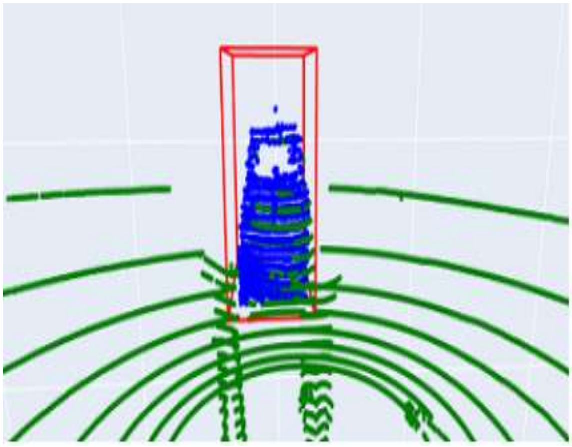

Fig.2: Blickfeld Cube Sensor 3D view for 171 frame ID

From the figures, it is observed that the greatest number of points inside 3D bounding box are for frame id 162 and 171. Figure 1 shows 3D view for 162 frame ID recorded over 2.45 meters, gives 2184 points inside 3D bounding box whereas for Figure 2, 2026 points of car are detected from the distance of 2.59 meters at 171 frame ID. This figure gives clear representation of car in both the frames.

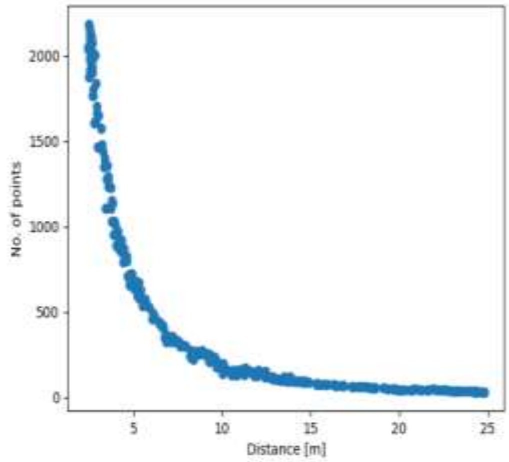

Fig.3: Graph of Number of Points using Blickfeld Cube Sensor Vs. Distance(m)

The graph provides information about Number of Points verses distance calculated inside 3D Bounding Box for Blickfeld Cube Sensor. In this scenario, Weplotted graph of 240 frames 5 which covered in 25m distance range. We can see that the number of points inside the bounding box increases as the car gets closer to the sensor, making it easy to identify the car. Less points recorded in the 3D Bounding box as the car moves away from the object. There are more points observe in between 2.82 and 3.88 meters. At distance of 24.8 meters, there are 31 points detected of the car which are minimum points view from overall recording. We can see from the graph that, distance and number of points of car are inversely proportional with each other. It conclude that the number of points are less as distance increases within the 3D bounding box.

B) Using the Velodyne Puck Sensor:-

Fig.4: Velodyne Sensor 3D view for 162 Frame ID

https://doi.org/10.46335/IJIES.2023.8.3.7 **e-ISSN: 2456-3463**
Vol. 8, No. 3, 2023, PP. 39-42

International Journal of Innovations in Engineering and Science, www.ijies.net

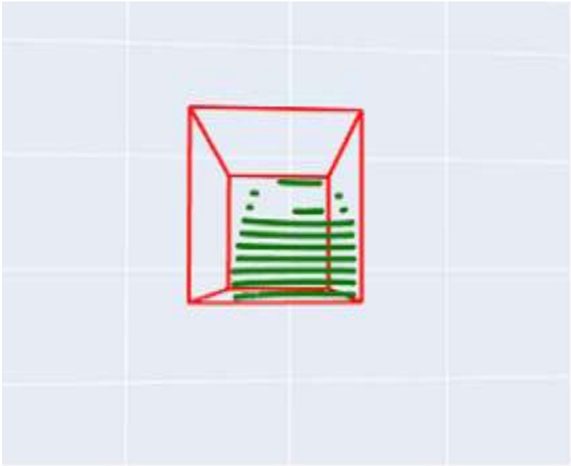

Fig.5: Velodyne Sensor 3D view for 153 Frame ID

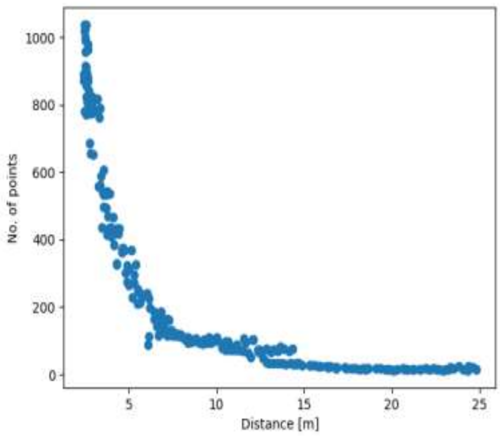

Fig.6: Graph of Number of Points using Velodyne Sensor Vs. Distance(m)

The figure above shows a Graph of Velodyne sensor distance Versus Number of points inside 3D bounding Box. There are fewer points inside 3D bounding box observed from the graph. As the car comes closer to the sensor, the number of points increases and the maximum points detected by sensor are nearly 1000 Points where as 31 points of the car observe inside the 3D bounding box when it is present at distance of 24.8 meters. There are more points between 2.82 and 3.99 meters are detected inside the 3D bounding box. The above graph shows that, the distance and number of points of the car are inversely proportional to each other, similar to the Blickfeld Cube sensor.

III. RESULT & DISCUSSION

Table: 1 Sensor wise frames, 3D bonding & distance

Sensor	Frame ID	Points in 3D Bounding Box	Distance(Metres)
Blickfeld Cube Sensor	5	35	24.05
	87	272	8.86
	162	2184	2.45
	167	1988	2.52
	193	923	4.15
	220	156	10.30
	239	95	14.67
Velodyne Puck Sensor	5	16	24.05
	87	101	8.86
	162	1018	2.45
	167	900	2.52
	193	384	2.52
	220	78	10.29
	239	33	14.67

The number of points within 3D bounding box of the two sensors are compared using random samples of frames. We may observe a significant difference in the points when we compare the calculated number of points within the 3D bounding box for both sensors for a specific frame number.[6] Compared to the Velodyne sensor point cloud, the Blickfeld Cube sensor point cloud produces a higher number of points.[5] The maximum number of points for Velodyne sensors inside the Boundary Box is 1018, while the maximum number of points for Blickfeld sensors is 2184, which is much larger than for Velodyne sensors at 162 frame ID. For each frame, the distance between the centre and the origin is nearly constant.

IV. CONCLUSION

The Blickfeld Cube sensor has a large object-detecting range than the Velodyne sensor and both sensors are used for autonomous driving. Compared to the Velodyne sensor, the Blickfeld Cube sensor's point cloud is extremely dense [4] We can identify the car more precisely with the Blickfeld Cube point cloud than with the Velodyne point cloud. The Blickfeld provide unique scan pattern. For both sensors, the maximum number of points inside the 3D bounding box is between 2 and 4 meters. It is evident that the number of points within the 3D bounding box for both sensors reduces as the distance rises, indicating that they are inversely proportional. The number of points within the 3D bounding box for the Blickfeld Cube sensor is higher because the vertical resolution is 4-500 lines scan per frame. Velodyne generates 360° field of view in Horizontal Resolution[2].

REFERENCES

[1] *https://elearning.rwu.de/pluginfile.php/396097/modresource/content/2/HandsOn.ipynb Accessed Date: November 01, 2022,*

[2] *https://en.wikipedia.org/wiki/VelodyneLidar*

[3] *https://arxiv.org/abs/1803.06184/Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P. & Yang, R. (2018)/The apolloscape dataset for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 954-960).*

[4] *Meinl, F., Stolz, M., Kunert, M., & Blume, H. (2017, March)/ An experimental high performance radar system for highly automated driving. In 2017MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM) (pp. 71-74).*

[5] *Kalman, R. E. (1960)/ A new approach to linear filtering and prediction problems. Journal of basic Engineering, 82(1), 35-45.*

[6] *Geiger, A., Lenz, P., & Urtasun, R. (2012, June)/ Are we ready for autonomous driving? the kitti vision benchmark suite/ In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 3354-3361).*