UNIVERSITY OF APPLIED SCIENCES RAVENSBURG-WEINGARTEN
LIDAR AND RADAR SYSTEMS

Task #03

Student name: *Akshay Pralhad Kanade*

Professor: *Dr. Stefan Elser*
Due date: *February 11th, 2022*

Introduction.

In order to estimate the next position of the car by assuming that the car is moving at the same speed, we collect data from the various sensors. All images are taken at the same time. In this dataset, we have access to data taken by a fisheye camera and two different types of lidars - Velodyne Puck and Blickfeld Cube. Two different situations were recorded with these sensors and the dataset was recorded. In the first situation (Record 1), the car comes directly toward the sensor and then moves back. The recordings for this situation were taken for 42 different time stamps. In the second situation (Record 2), the car drives parallel to the sensor set. The recordings for this situation were taken for 14 different timestamps. From these two data sets, we need to predict the future position of the car using the bounding box. Then the ground truth of the 3D bounding box and predicated 3D bounding box is compared and the error between them is calculated.

Algorithm.

First create a 3D bounding box by changing the mode type in the program to "lines + markers". This will give you a clear idea of the 3D bounding box. Kalman filters are very popular in most cases of state estimation, by taking the midpoints of two consecutive indices and predicting the midpoints of the next index. The midpoints of an index consists of the x, y and z coordinates and the width(w), height(h), length(l), and yaw of the vehicle. To calculate the velocity of the vehicle, the value of time was assumed to be constant, and estimated the velocity of the vehicle along the three dimensions using matrix multiplication. The covariance matrix is our uncertainty matrix in the estimation. Then, this matrix was used to calculate the midpoint of the car position. The new midpoint consists of $x_{k+1} = (x, y, z, w, l, h, yaw, v_x, v_y, v_z, deltayaw)$.

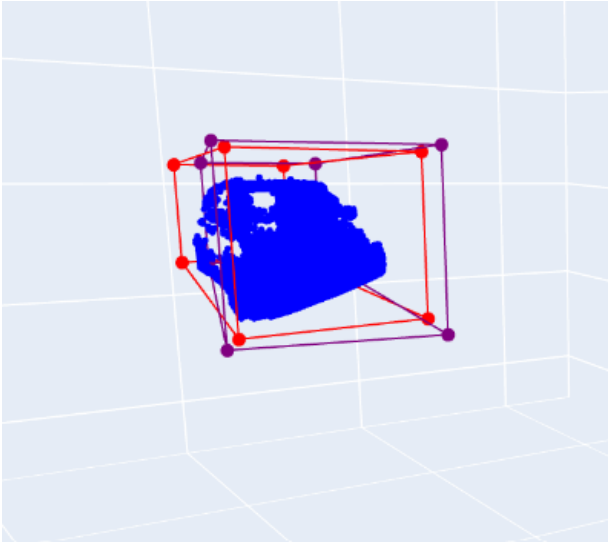
Draw the 3D bounding box using the new calculated center point. Compare the actual and the predicted center of the 3D bounding box and calculate the percent error.The equation for the total percentage error is $(x_{error} + y_{error} + z_{error} + ((0.2) * yaw_{error}))/4$.

In the above equation, as the error is predicted and the actual values of x, y, and z are almost small and identical, the multiplication factor 1 was chosen. But, as the error value of yaw is higher, 0.2 was used as the scaling factor for yaw error.

Evaluating the dataset "Record 1".

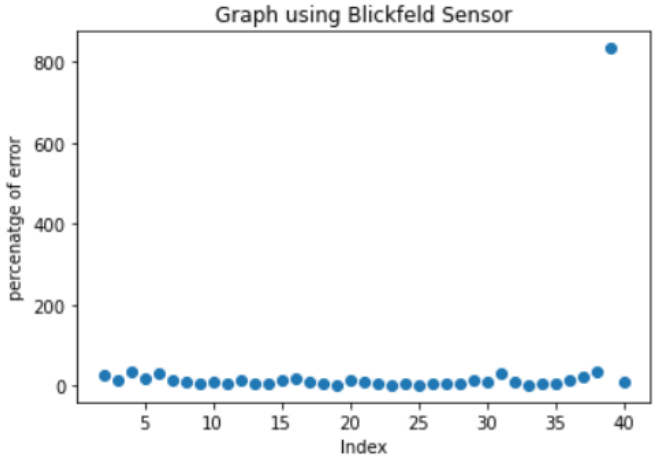
(The car moves forward/backward from the sensor)-.

Using the Blickfeld Cube sensor –.



As you can see from the images, I have eliminated the points of the Blickfeld Cube sensor that are outside the 3D bounding box using an algorithm. There are two 3D bounding boxes, one with red color and one with purple color. The red 3D bounding box is an actual 3D bounding box, while the purple 3D bounding box is an assumed 3D bounding box. The picture of index 20 where the difference between the actual and the predicted 3D bounding box can be clearly seen.

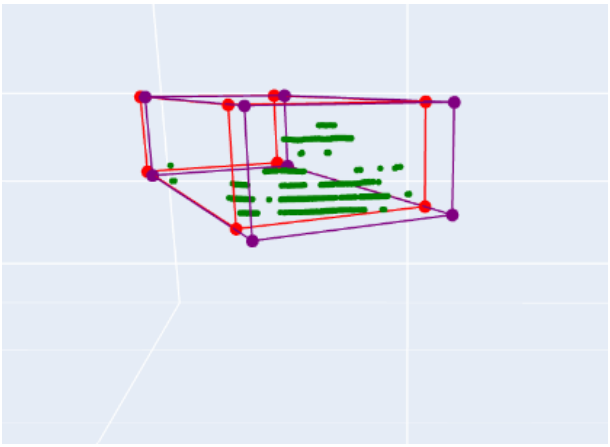
Graph plotted using Blickfeld Cube sensor datapoints.



The figure above shows the percent error versus the index for the Blickfeld Cube sensor. The percent error between the actual and predicted 3D bounding box for each index was calculated. It can be clearly seen that the percent error is almost the

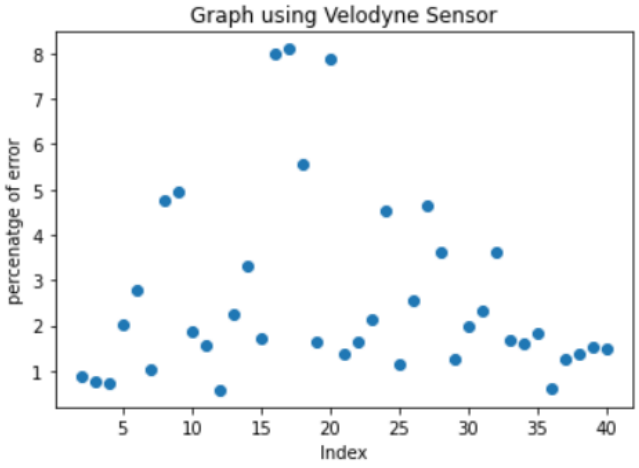
same for all the indices except for the index number 39, where the percent error value ranges from 1 to 25%. The error percentage for the 39th index is 833% because of the large change in the yaw value. The equation for the total error works fine for almost all indexes.

Using the Velodyne sensor –.



As you can see from the images, an algorithm to eliminate the points on the Velodyne sensor that are outside the 3D bounding box was used. There are two 3D bounding boxes, one with red color and one with purple color. The red 3D bounding box is an actual 3D bounding box, while the purple 3D bounding box is an assumed 3D bounding box. I have taken a picture of Index 18 where the difference between the actual and the predicted 3D bounding box can be clearly seen.

Graph plotted using Velodyne sensor datapoints.

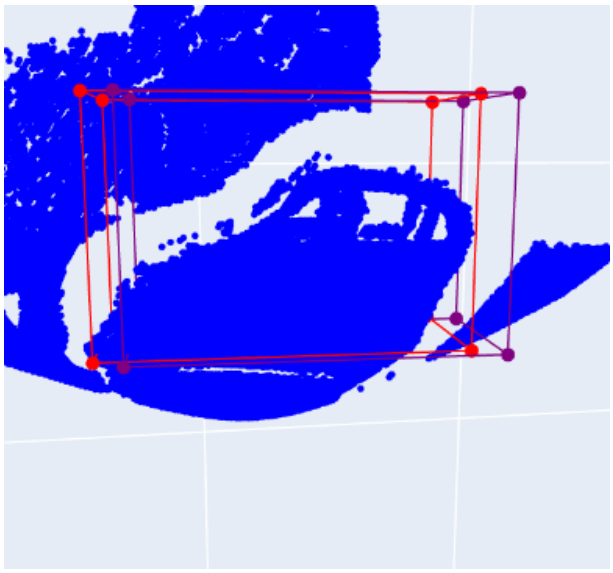


The figure above shows the percent error versus index for the Velodyne sensor. The percent error between the actual and predicted 3D bounding box for each index was calculated. It can be clearly seen that the percent error is small for all indexes. The percent error value is between 0 and 8%. The maximum percent error is for index 18 which is 8.09%. There is no particular pattern in the graph, but the percent error values are too small for the Velodyne sensor.

Evaluating the dataset "Record 2".

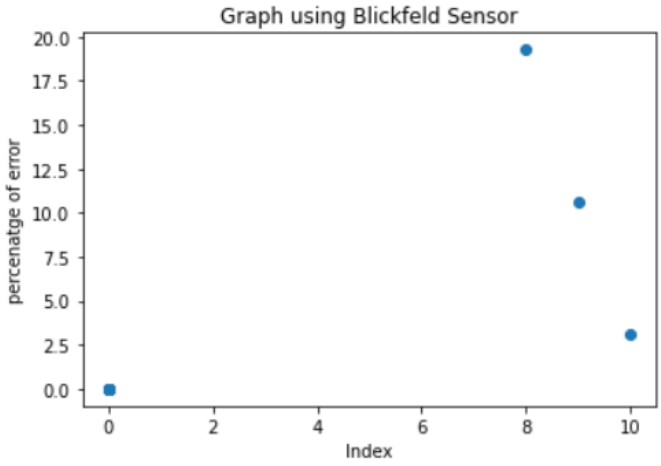
(The car moves parallel to the sensor)-.

Using the Blickfeld Cube sensor –.



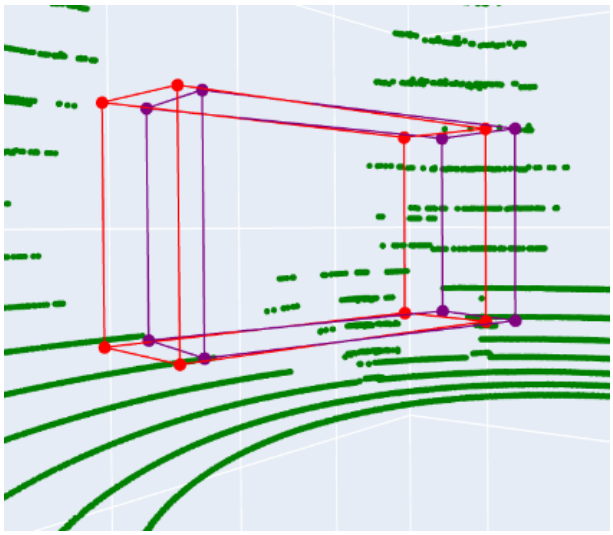
As you can see in the pictures, the car moves parallel to the sensor. There are two 3D bounding boxes, one with red color and one with purple color. The red 3D bounding box is an actual 3D bounding box, while the purple 3D bounding box is an assumed 3D bounding box. There are 14 indexes available, but of the 14 indexes, center points are only available for three indexes. The picture of index 8 where the difference between the actual and the predicted 3D bounding box can be clearly seen.

Graph plotted using Blickfeld Cube sensor datapoints.



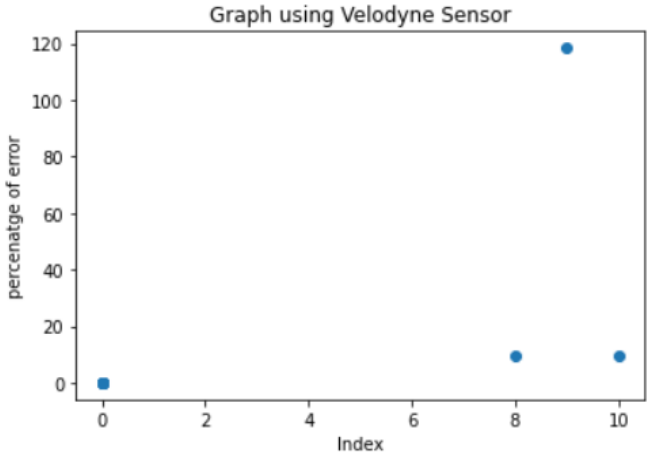
The figure above shows the percent error versus index for the Blickfeld Cube sensor. The percent error between the actual and predicted 3D bounding box for each index was calculated. There is a percent error available for three indexes. The percent error for indexes 8 and 9 is 19.27% and 10.59%, respectively. For index 10, the percentage error is 3.12%.

Using the Velodyne sensor –.



Similar to how the Blickfiled Cube sensor has only three indices, Velodyne also has only three indices. As you can see in the pictures, there are two 3D bounding boxes, one with red color and one with purple color. The red 3D bounding box is an actual 3D bounding box, while the purple 3D bounding box is an assumed 3D bounding box. The picture of Index 8 where the difference between the actual and assumed 3D bounding box can be clearly seen.

Graph plotted using Velodyne sensor datapoints.



The figure above shows the percent error compared to the index for the Velodyne sensor. The percent error between the actual and predicted 3D bounding box for each index was calculated. There is a percent error available for three indexes. The percent error for indexes 8 and 10 is 9.80% and 9.70%, respectively. For index 9, the percent error is 118.36% because the yaw value of the actual versus the predicted 3D bounding box has changed significantly.

Comparison of results.

For Record 1-.

Sensors	Index Number	Error Percentage
Blickfeld Cube	2	25.97%
	23	1.97%
	31	33.07%
	39	833.16%
Velodyne Puck	2	0.90%
	17	8.09%
	23	2.12%
	39	1.51%

For Record 1, if we compare the calculated percent error for both sensors for a given index number, we can see a large difference in the percent error. The percent error of the Blickfeld Cube sensor point cloud is higher than that of the Velodyne sensor point cloud. For index 39, the percent error of the Blickfeld Cube sensor is 833.16%, while the percent error of the Velodyne sensor is 1.51%.

For Record 2-.

Sensors	Index Number	Error Percentage
Blickfeld Cube	8	19.27%
	9	10.59%
	10	3.12%
Velodyne Puck	8	9.80%
	9	118.36%
	10	9.70%

For Record 2, if we compare the calculated percent errors of both sensors for a given index number, we can see a large difference in the percent error. The percent error of the Blickfeld Cube sensor point cloud is similar to that of the Velodyne sensor point cloud. At index 9, there is a large difference in the percent error due to a large change in yaw values. The percent error of the Blickfeld cube sensor is 10.59%, while the percent error of the Velodyne sensor is 118.36%.

Conclusion.

The error percentage of the Blickfeld Cube sensor is higher than that of the Velodyne sensor. The 3D bounding box calculated by the Velodyne sensor is almost equal to the actual 3D bounding box. For some indexes, there is a large percentage error due to the large change in yaw values between two consecutive indexes. Using the point cloud of the Velodyne sensor, we can determine the next position of the vehicle more accurately than using the Blickfeld Cube sensor.

References.

1. Felix Berens, Saravanan. Real time Autonomous dataset with blickfield and Velodyne LIDAR sensors 2021

2. Rusinkiewicz, Szymon; Marc Levoy (2001). Efficient Variants of the ICP Algorithm. Proceedings Third International Conference on 3-D Digital Imaging and Modeling. Quebec City, Quebec, Canada.

3. Rövid et al. "Raw fusion of camera and sparse LiDAR for detecting distant objects", at 2020