XV-11 Laser Distance Sensor Feasibility Analysis Report

by Jiaxin Li

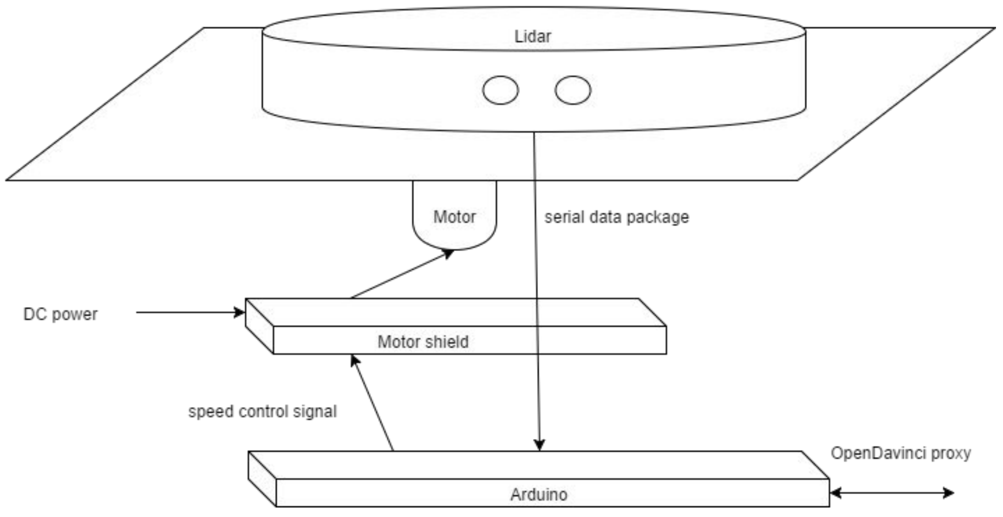
Brief

The lidar we intend to use on the car originally comes from Neato Robot Vacuum Cleaner, with NO official document available online. The major data source of this report comes from online forum and by experiment.

The XV-11 laser distance sensor (abbrev. Lidar in the context) is designed to work with an digital signal processor customized for the robot, however, since the data interface is standard serial input, the data might be adjusted to suit the application.The lidar is supposed to work within the distance of 6 meters with ideal sample rate of 300 r.p.m (5 frame per second)and resolution of 1 reading per degree, accuracy drops with increasing distance or due to nonideal surface.

Overall structure

Currently we are using an Arduino Mega board to drive the Lidar(the reasoning will be provide afterwards). Later on, we might replace the arduino board with a more compact motor driver board for the simplicity reason.



Serial Data Structure

A full revolution will yield 90 packets, containing 4 consecutive readings each.
The length of a packet is 22 bytes. This amounts to a total of 360 readings (1 per degree) on 1980 bytes.
Each packet is organized as follows:
<start byte> <index> <speed> <Data 0> <Data 1> <Data 2> <Data 3> <checksum>
where:
<start byte> is always 0xFA
<index> is the index byte in the 90 packets,
going from A0 (packet 0, readings 0 to 3) to F9 (packet 89, readings 356 to 359).

<speed> is a two-byte information, little-endian. It represents the speed.
<Data 0> to **<Data 3>** are the 4 readings. Each one is 4 bytes long, organized as follows :

byte 0 : <distance 7:0>

byte 1 : <"invalid data" flag> <"strength warning" flag> <distance 13:8>

The bit 7 of byte 1 seems to indicate that the distance could not be calculated. when this bit is set, the second byte is always 80, and the values of the first byte seem to be only 02, 03, 21, 25, 35 or 50... When it's 21, then the whole block is 21 80 XX XX, but for all the other values it's the data block is YY 80 00 00 maybe it's a code to say what type of error (35 is preponderant, 21 seems to be when the beam is interrupted by the supports of the cover)

byte 2 : <signal strength 7:0>

byte 3 : <signal strength 15:8>

Byte 2 and 3 are the LSB and MSB of the strength indication. This value can get very high when facing a retroreflector.

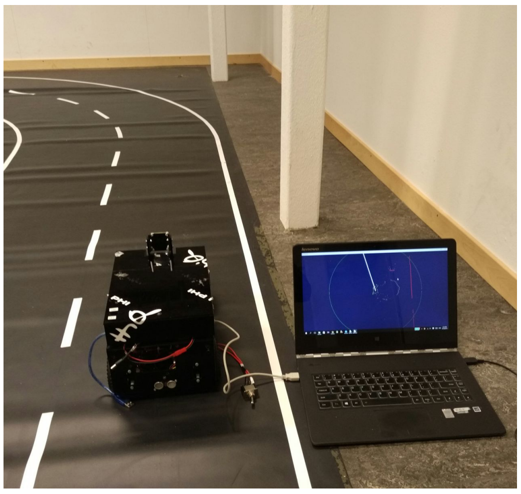
<checksum>

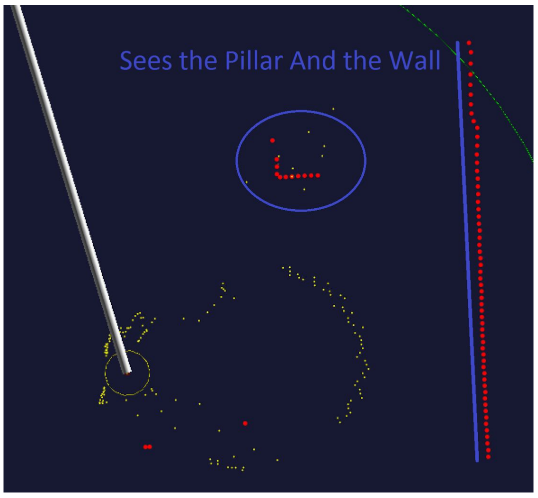
checksum is a two-byte checksum of the packet, provided that data is the list of the 20 first bytes, in the same order they arrived in.

currently the arduino streaming the data packet at baud rate of 115200 (bit/second), which could handle 7 packet (15840bit/packet) per second in theory.And also get the information about the measurement “quality” indicating if the data is trustable.

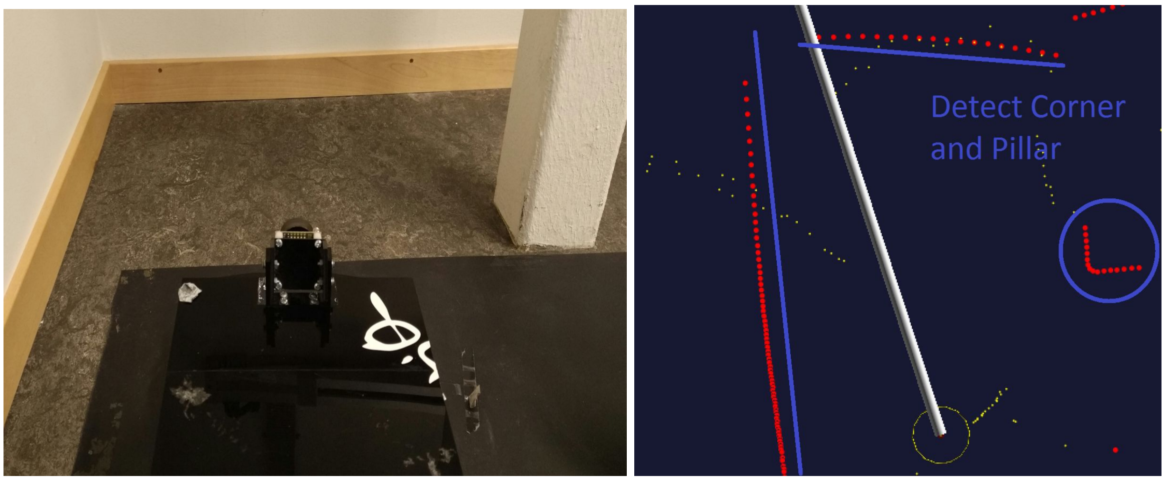
Data Visualization & example Data

For the purpose of having a better understanding of the data collection, and for a more obviously and intuitively view, a program for visualize the data was introduced at beginning. And some test was carried on and result was collected and visualized (scenario on the left and result with comment on the right).

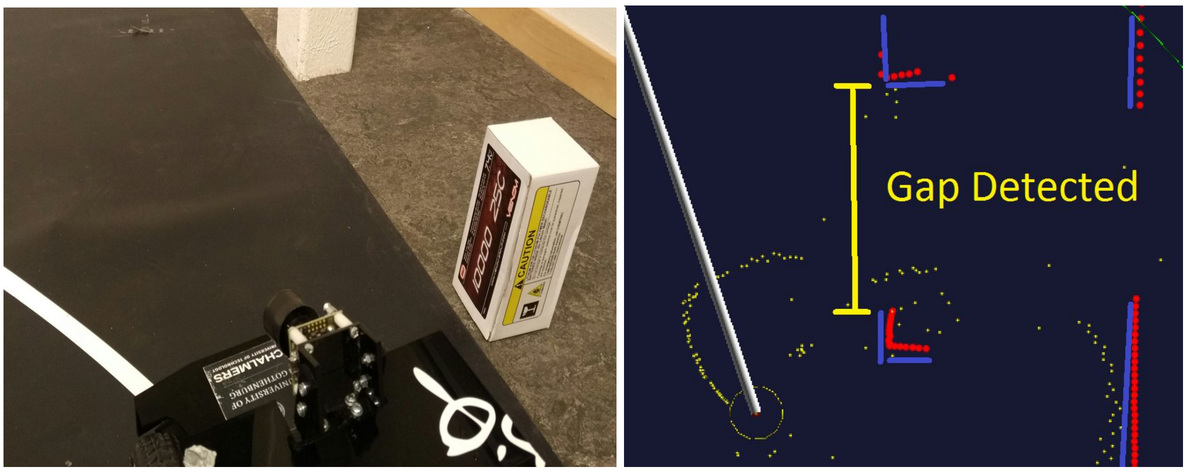




Intend for Roadside Object Detection



Intend to test the ability to detect the angle as a pattern



Intend to test the ability to detect the Parking Gap.

Intention of using the data

The main intention for using the Lidar data is for improve the parking algorithm. As the most beneficial feature for using the Lidar data is will provide the parking slot information more precisely and will provide a global view for the surrounding environment if following some advanced algorithm.

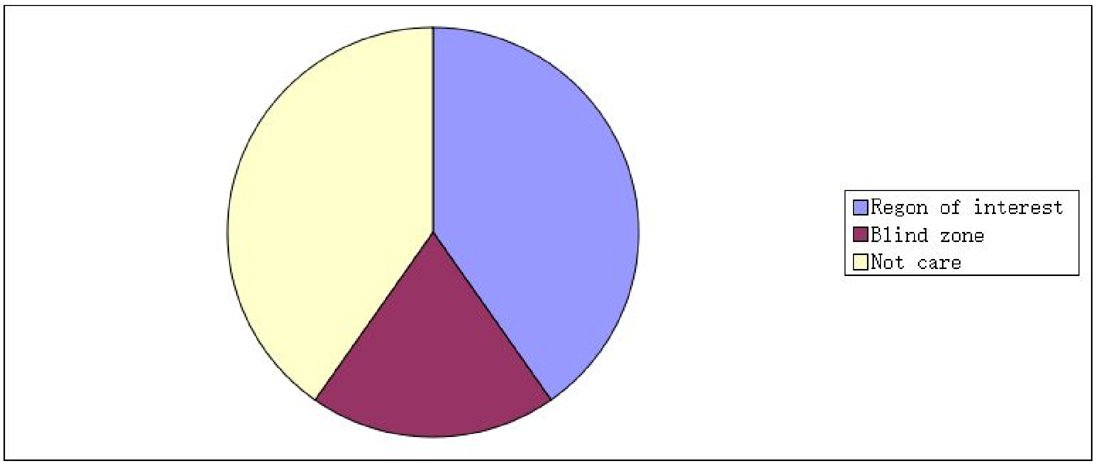
The basic idea (or the rough algorithm we came up for now)for how to use the data is as follows:

Decode the packet:

the data will be presented as polar coordinates form which could be simplified as series of **[Direction, Distance, Quality]** records in the conceptional level.

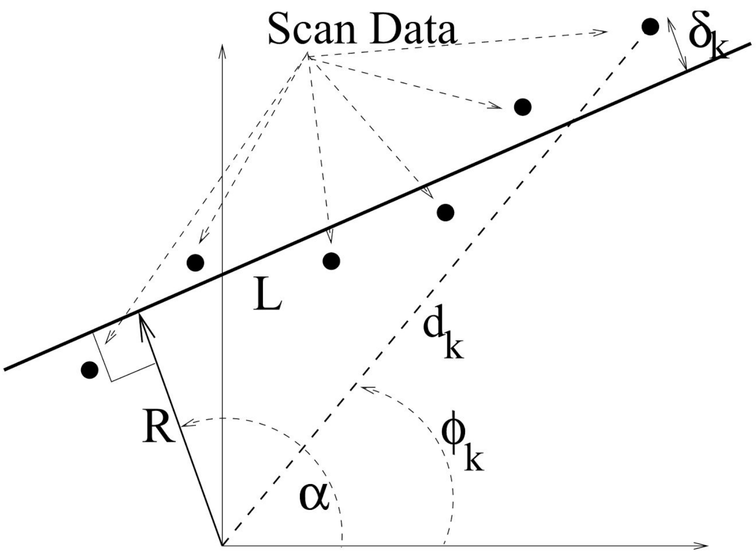
Filtering the Region of interest:

Since we are aiming at the goal of finding the parking slot and assist parking algorithm, not all decoded data contains useful information. Thus, some filter criteria will be introduced to narrow down the region of interest. And computational task intensity could also be decreased.

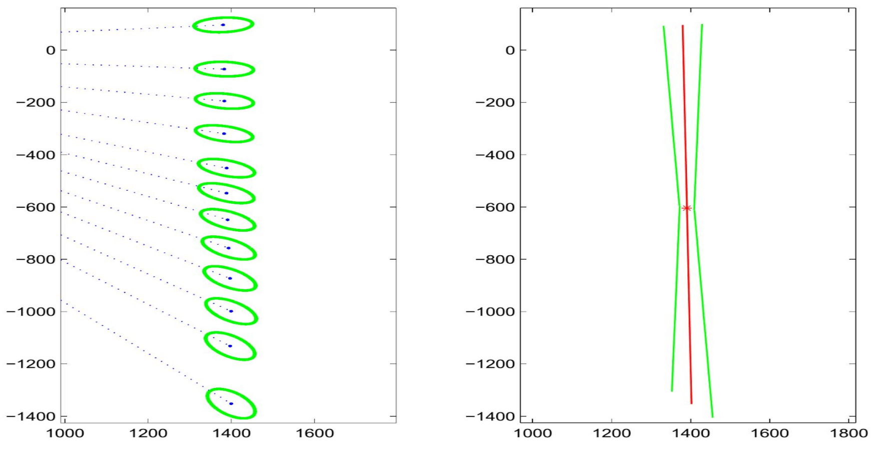


Line fitting:

A possible solution for finding out the line is transform the input data using the **Weighted Line Fitting algorithm.** (Hough Transform might be used)

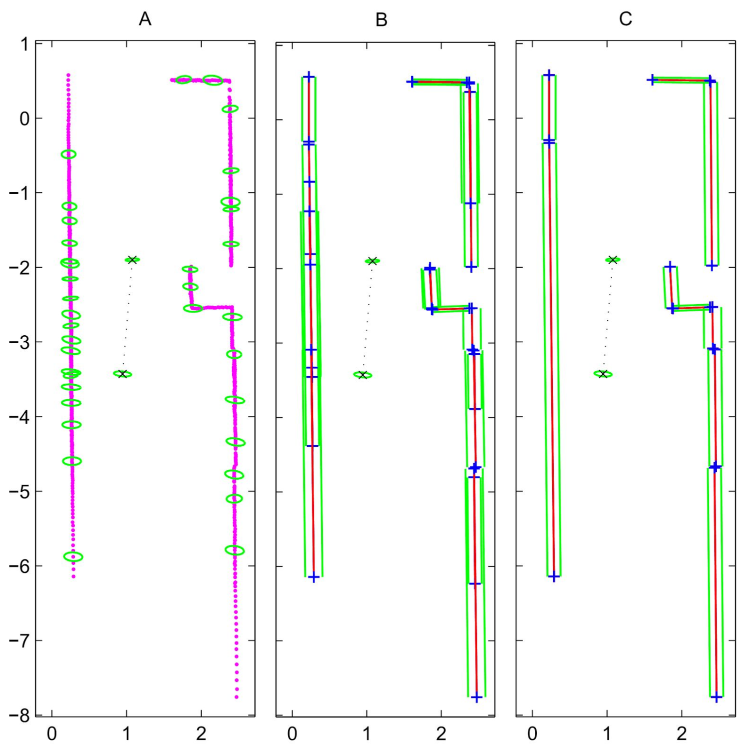


And below is an example of line segment fit: data points (left) and fitted line with a representation of its uncertainty.



Pattern Matching:

features from the result of line fitting(angle and length) could be used to matching a certain pattern described in parking scenario.



An ideal scenario we could expect after the execution of the algorithm is the red dot line in the fig.