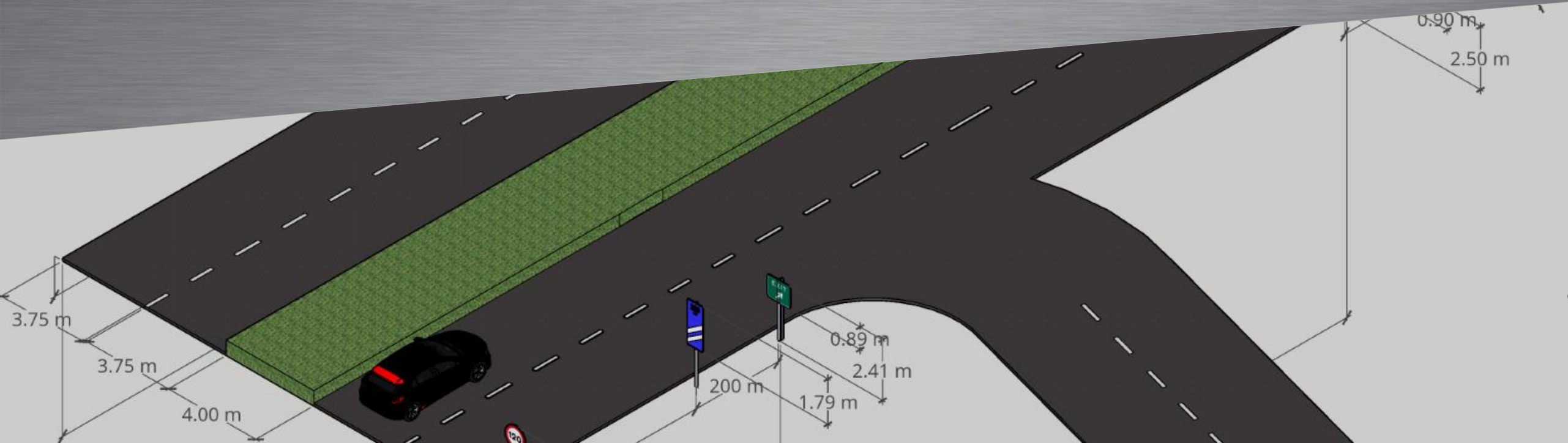
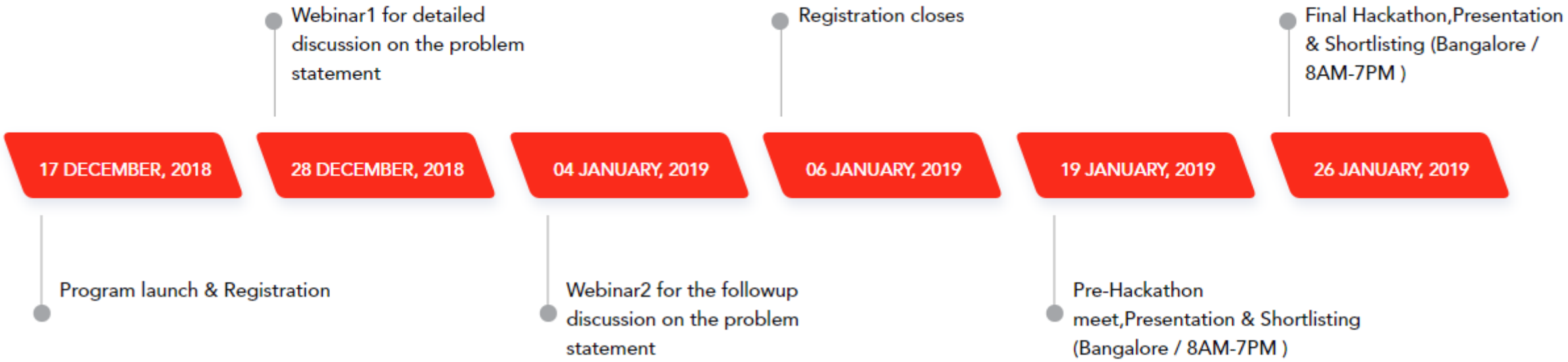
DAIMLER

MBRDI CEA Virtual Drive Challenge



MBRDI CEA Virtual Drive Challenge

1. Knowledge, skill & creativity contributing to the next chapter in Human mobility!
2. Create a concept and prototype to simulate LIDAR data, in response to a car driving in a virtual road environment



Basics

Light Detection and Ranging (LIDAR)

LIDAR transmits pulses of laser light and creates a high spatial density map of the surrounding

Theory

- A pulse of light is emitted and time is recorded
- The reflection of that pulse is detected and the time is recorded
- Using speed of light, the delay can be converted into distance
- Using the position and orientation of the LIDAR, coordinate of the reflective surface can be computed in Cartesian frame

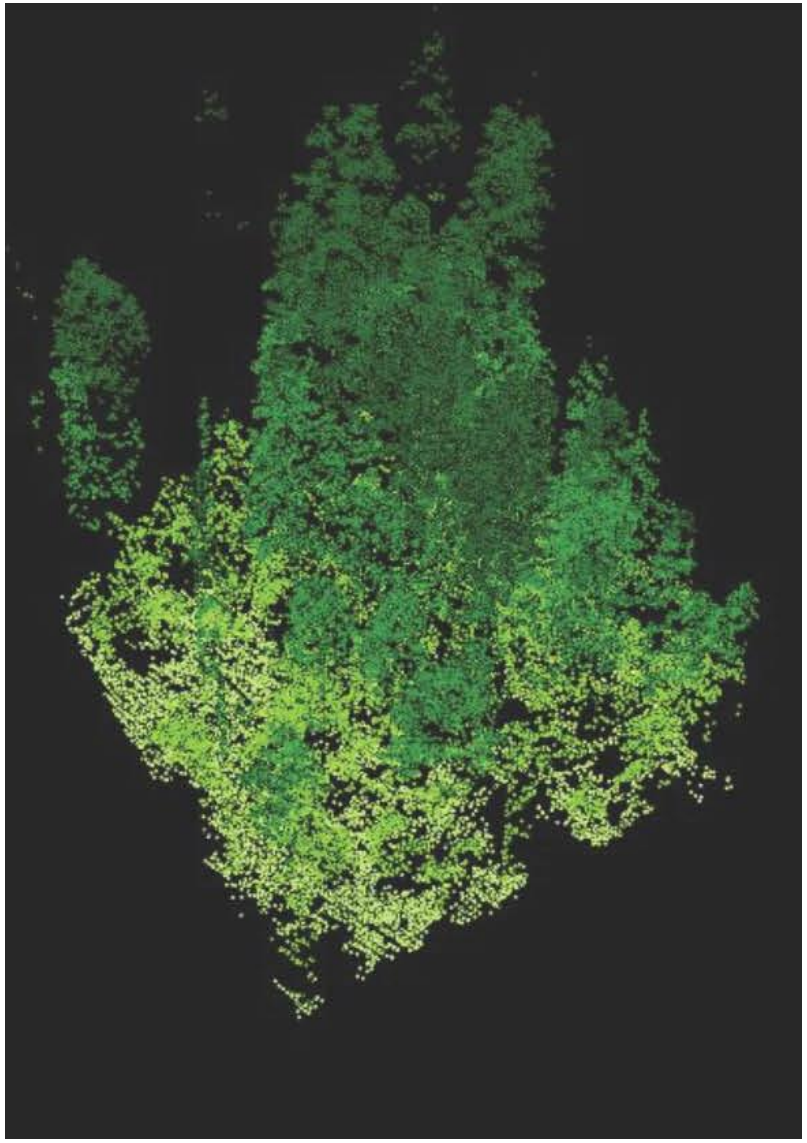
How LIDAR sees the world?

- Velodyne LIDAR: Overpasses – Explains how the LIDAR represents the cloud point of an overpass

https://www.youtube.com/watch?v=oZ7P4RsTE64

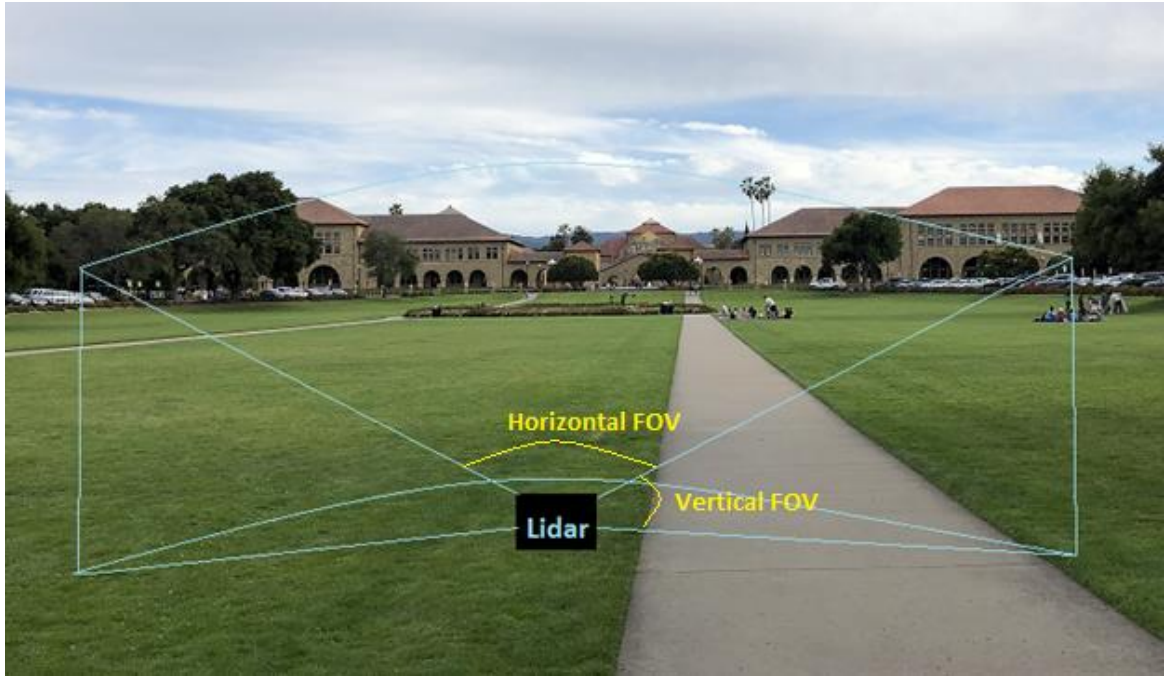
- Velodyne LIDAR VLS-128™ Sensor

https://www.youtube.com/watch?v=KxWrWPpSE8I

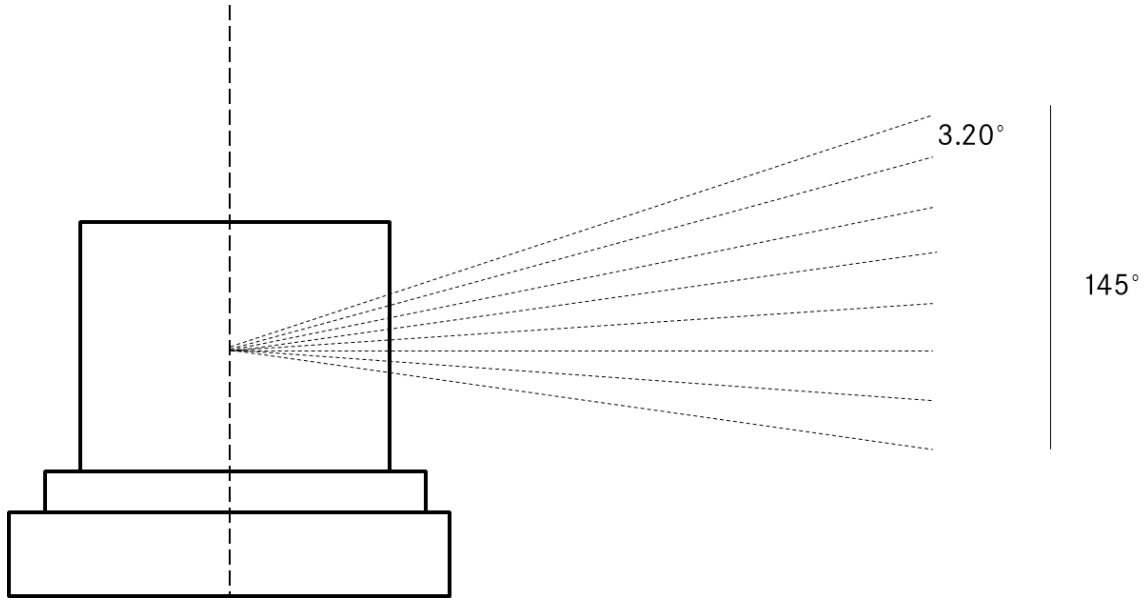


Field of View (FOV)

Field of view (FOV) is the extent of the observable world that is seen at any given moment. In the case of optical instruments or sensors it is a solid angle through which a detector is sensitive to electromagnetic radiation



Graphical depiction of field of view concept



Representation of an 8 channel LIDAR

LIDAR Mount on vehicle

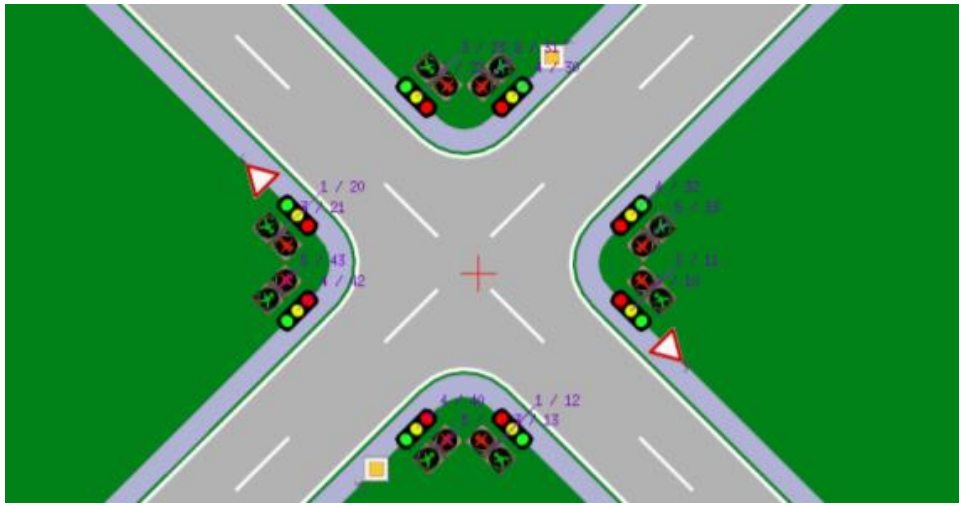
- LIDAR mounts are usually on roof rails to take advantage of 360 degree view reducing occlusion of vehicle parts.

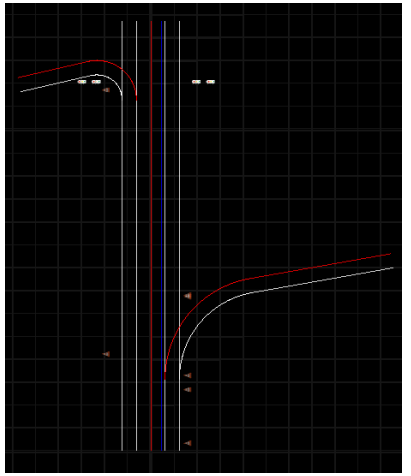
- LIDAR mount has to be configurable in the software, which gives it the capability to build a scalable model for point cloud generation. LIDAR's position in software to be provided in a X,Y,Z Cartesian frame. Vehicle grill center is considered to be X=0, Y=0

Virtual representation of roads

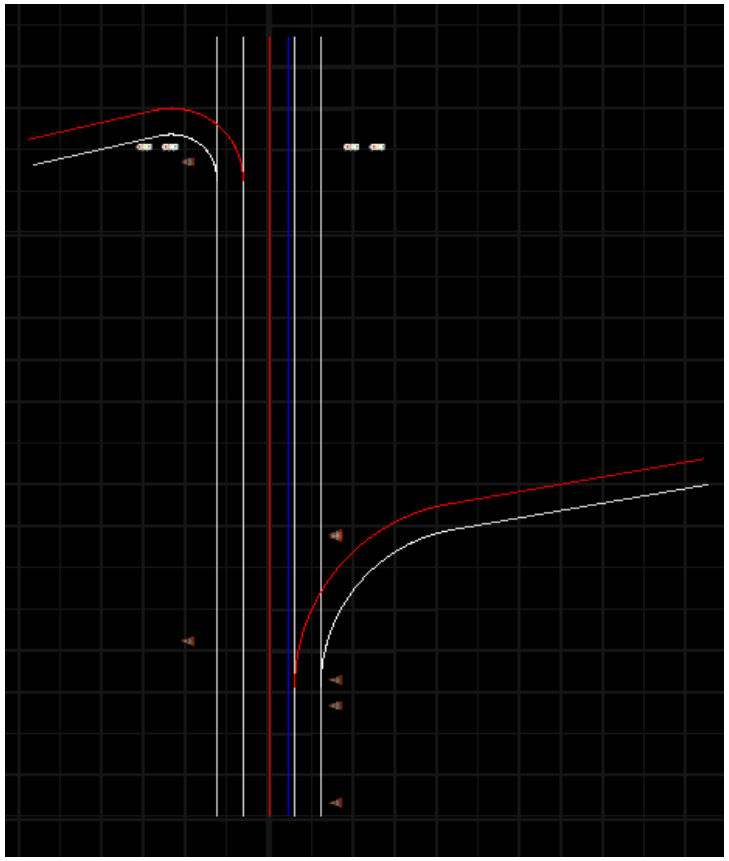
- OpenDRIVE is an open format specification to describe a road network's logic.
- OpenDRIVE files are designed to describe entire road networks with respect to all data belonging to the road environment. Entities interacting with the road is not described in OpenDRIVE format
- Road geometry is represented standard mathematical equations.
- Specification of OpenDRIVE is available in internet.
- Output of OpenDRIVE is in XODR format, schema of the format is readily available and can be parsed

http://www.opendrive.org/docs/OpenDRIVEFormatSpecRev1.4H.pdf





OpenDRIVE schema



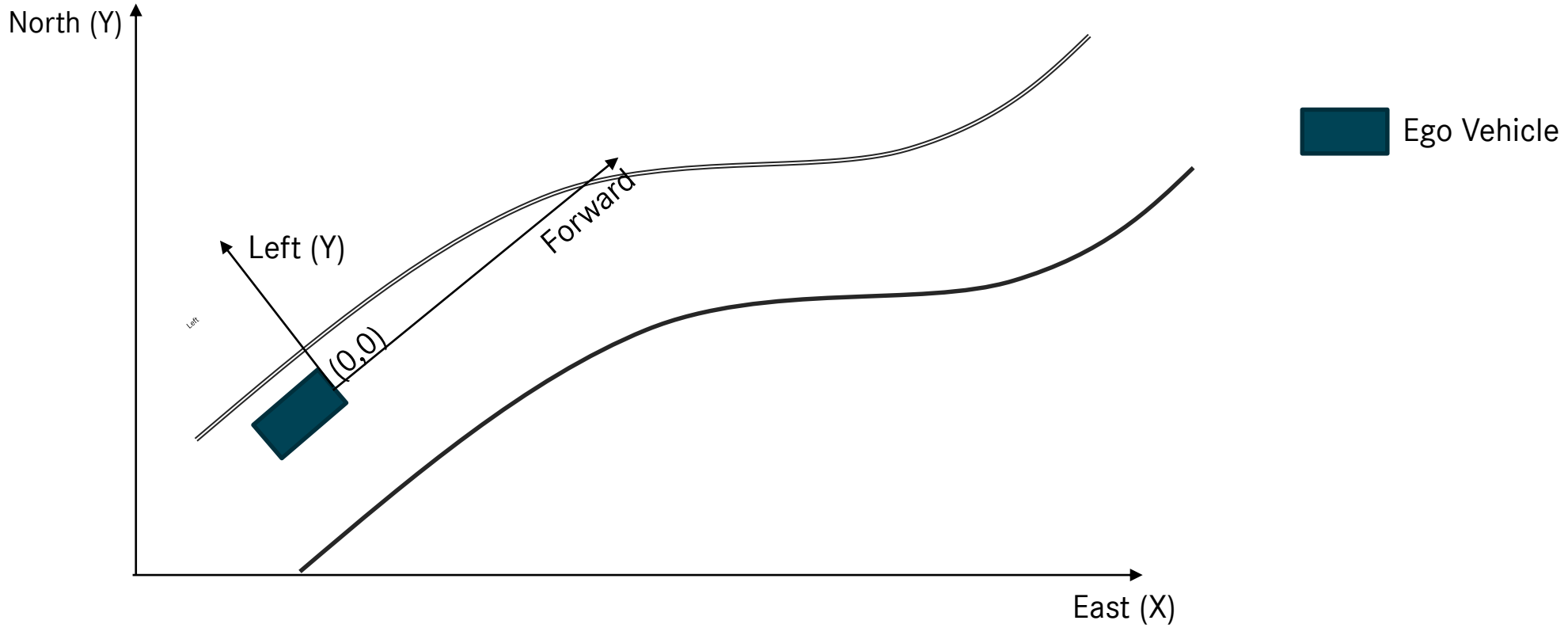
Road geometry modelled in OpenDrive format

bead name	level	instances per parent	parent	chapter
OpenDRIVE	0	1	-	5.1
|-header	1	1	OpenDRIVE	5.2
| |-geoReference	2	0..1	header	5.2.1
|-road	1	1+	OpenDRIVE	5.3.1
| |-link	2	0..1	road	5.3.2
| | |-predecessor	3	0..1	link	5.3.2.1
| | |-successor	3	0..1	link	5.3.2.2
| | |-neighbor	3	0..2	link	5.3.2.3
| |-type	2	0+	road	5.3.3
| | |-speed	3	0+	type	5.3.3.1
| |-planView	2	1	road	5.3.4
| | |-geometry	3	1+	planview	5.3.4.1
| | | |-line	4	1	geometry	5.3.4.1.1
| | | |-spiral	4	1	geometry	5.3.4.1.2
| | | |-arc	4	1	geometry	5.3.4.1.3
| | | |-poly3	4	1	geometry	5.3.4.1.4
| | | |-paramPoly3	4	1	geometry	5.3.4.1.5
| |-elevationProfile	2	0..1	road	5.3.5
| | |-elevation	3	0+	elevationProfile	5.3.5.1
| |-lateralProfile	2	0..1	road	5.3.6
| | |-superelevation	3	0+	lateralProfile	5.3.6.1
| | |-crossfall	3	0+	lateralProfile	5.3.6.2
| | |-shape	3	0+	lateralProfile	5.3.6.3
| |-lanes	2	1	road	5.3.7
| | |-laneOffset	3	0+	lanes	5.3.7.1
| | |-laneSection	3	1+	lanes	5.3.7.2
| | | |-left	4	0..1	lane section	5.3.7.2.1
| | | | |-lane	5	1+	left	5.3.7.2.1.1
| | | | | |-link	6	0..1	lane	5.3.7.2.1.1.1
| | | | | | |-predecessor	7	0..1	link	5.3.7.2.1.1.1.1
| | | | | | |-successor	7	0..1	link	5.3.7.2.1.1.1.2

Coordinate system

OpenDRIVE format allows the position of elements on the road to be defined using two sets of X, Y, Z coordinates

- Position of object w.r.t. position of vehicle
- Position of object w.r.t. a global coordinate origin



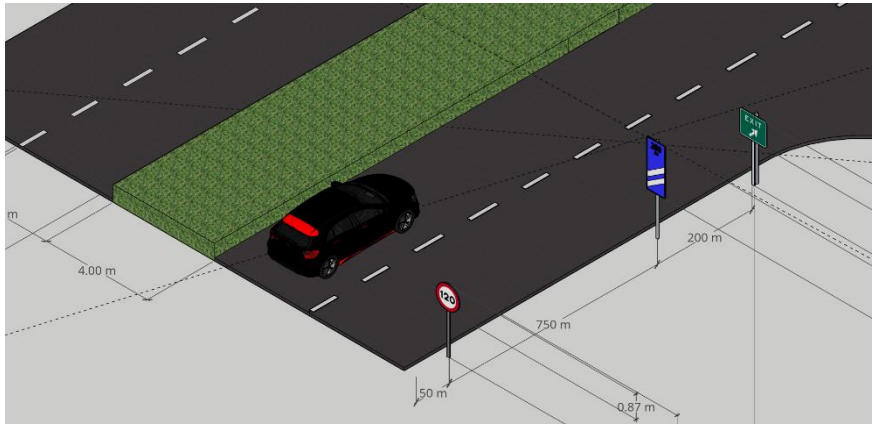
Problem statement

Problem statement - Overview

1. Generate simulated LIDAR point cloud data for any 3D object on the road, i.e. traffic signs, traffic signals, lane markings, road side barriers
2. Generated point cloud should take into account the field of view of the LIDAR sensor and placement of the sensor on the vehicle

Inputs provided

1. Conceptual overview document explaining how the sensor is fitted on the car, the working principle and output
2. Input and Output format specification, template code explaining calling and output conventions
3. Virtual road 3D model, specification in openDRIVE format



Online submission

As part of their submissions, participants are expected to include the following:

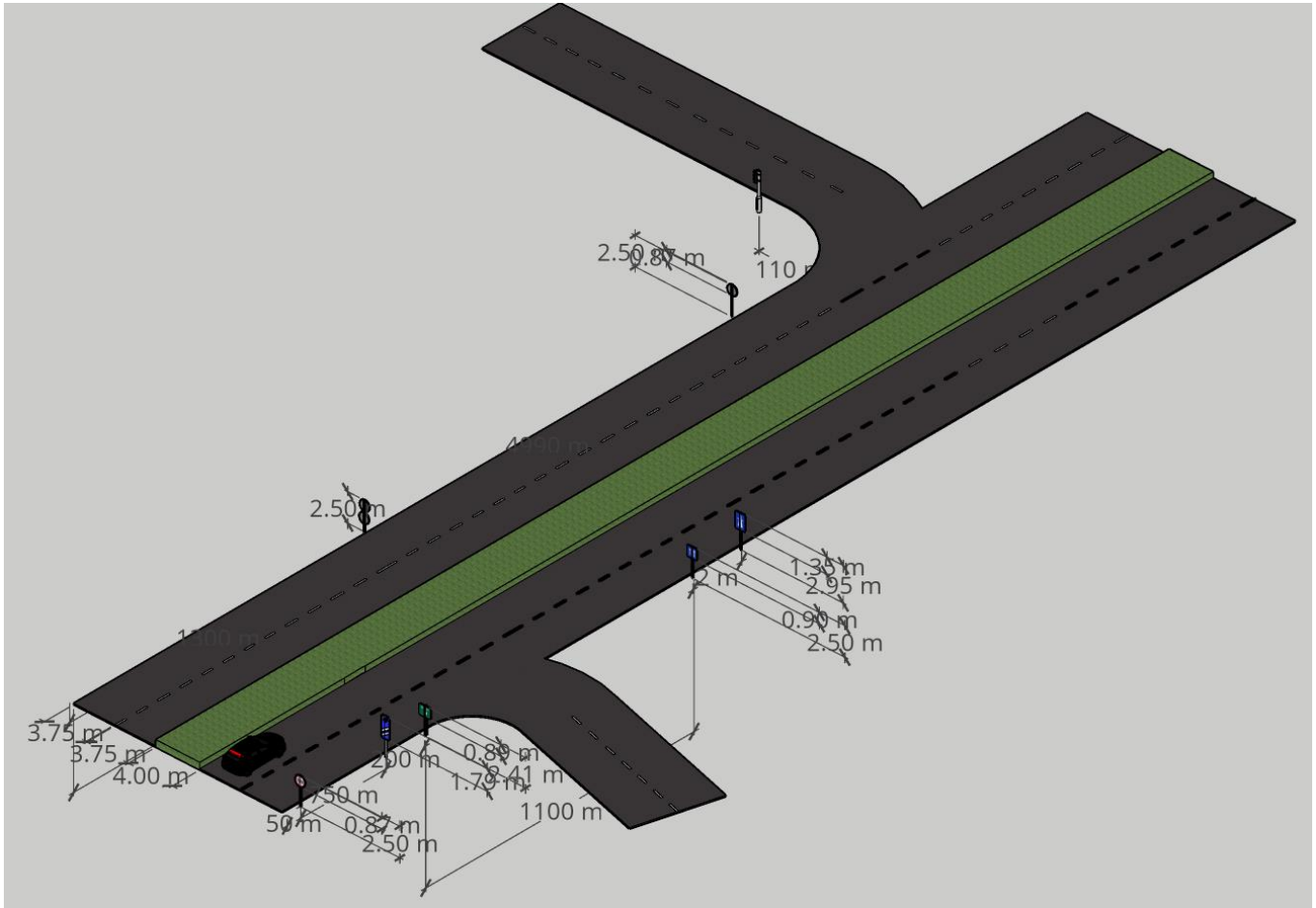
1. A write-up explaining the approach they've used to solve the problem
2. An algorithm/pseudo code explaining the simulation steps

Hackathon

1. A complete implementation of a software library that can generate point cloud data based on inputs provided

Virtual road to be used

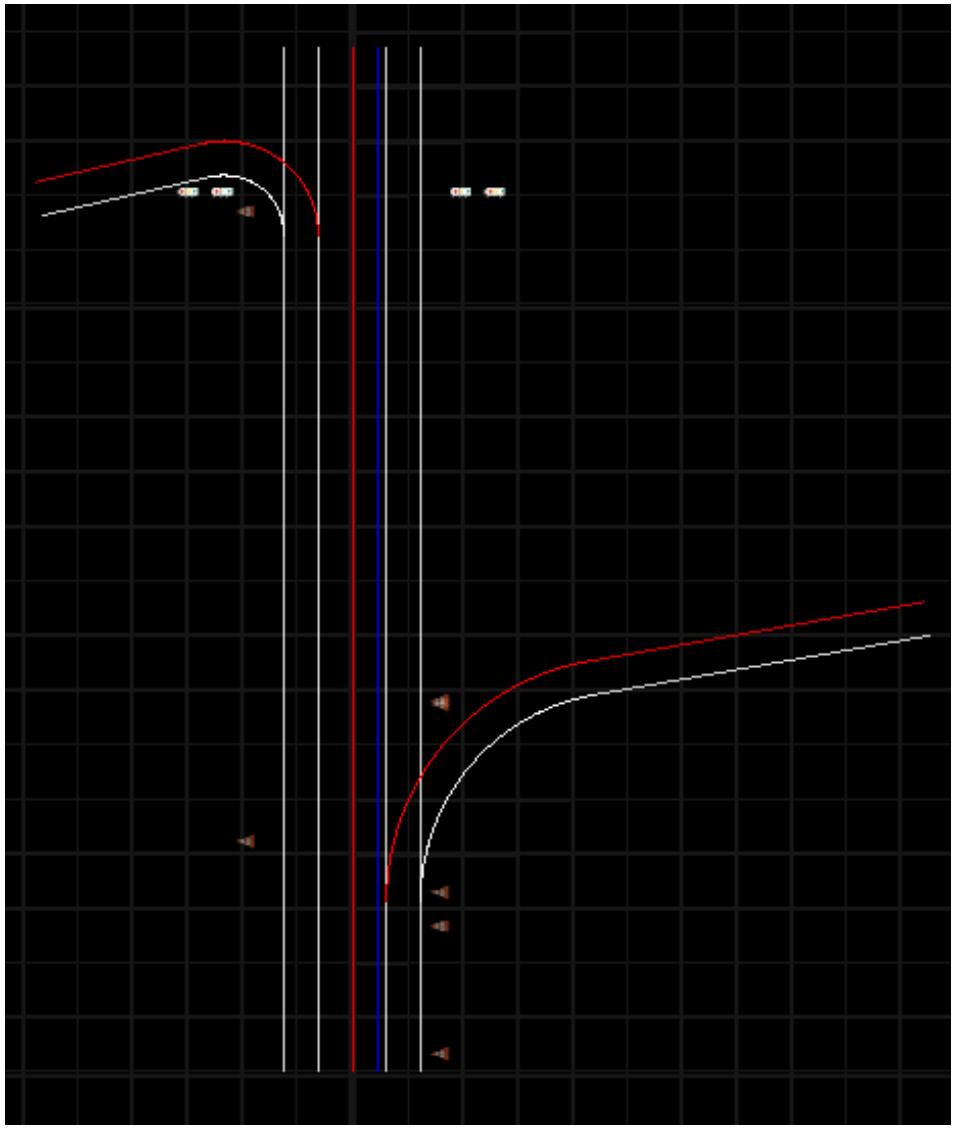
- Virtual road is a 4 lane, two-way highway, with one exit and entry points
- 7 km in length – dimension details are modelled in SketchUp web 3D model software



Instructions to view the .skp file:

1. Download and install SketchUp viewer for desktop
2. Open the file road_specification_v1.skp in the viewer
3. Use the 'measuring tape' tool to measure dimensions in the model

OpenDRIVE file to be used



```
<xs:sequence>
    <xs:element name="laneOffset" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="userData" type="userData" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="include"  type="include"  minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="s"    type="xs:double"/>
            <xs:attribute name="a"    type="xs:double"/>
            <xs:attribute name="b"    type="xs:double"/>
            <xs:attribute name="c"    type="xs:double"/>
            <xs:attribute name="d"    type="xs:double"/>
        </xs:complexType>
    </xs:element> <!-- end laneOffset -->
    <xs:element name="laneSection" maxOccurs="unbounded">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="left" minOccurs="0" maxOccurs="1">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="lane"     type="lane"     maxOccurs="unbounded"/>
                            <xs:element name="userData" type="userData" minOccurs="0" maxOccurs="unbounded"/>
                            <xs:element name="include"  type="include"  minOccurs="0" maxOccurs="unbounded"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="center" minOccurs="1" maxOccurs="1">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="lane"     type="centerLane" minOccurs="0" maxOccurs="1"/>
                            <xs:element name="userData" type="userData"   minOccurs="0" maxOccurs="unbounded"/>
                            <xs:element name="include"  type="include"    minOccurs="0" maxOccurs="unbounded"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="right" minOccurs="0" maxOccurs="1">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="lane"     type="lane"     maxOccurs="unbounded"/>
                            <xs:element name="userData" type="userData" minOccurs="0" maxOccurs="unbounded"/>
                            <xs:element name="include"  type="include"  minOccurs="0" maxOccurs="unbounded"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="userData" type="userData" minOccurs="0" maxOccurs="unbounded"/>
```

Specification: Field of view (FOV)

In the point cloud simulation library, the FOV should be configured using given parameters

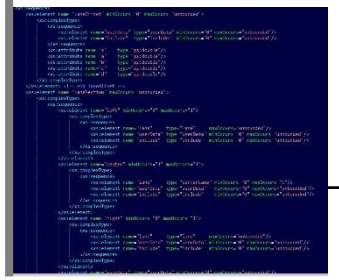
- Horizontal FOV : 145+° (azimuth)
- Vertical FOV : +15° to -25°(Elevation)
- Assume a 64 channel LIDAR, each channel is separated by 0.625°

Programmatic model

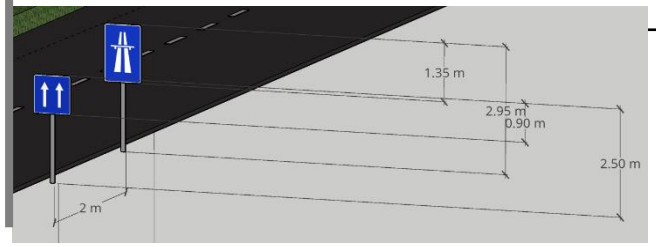
1. Software library should use a standard API in the following format:

```
struct Pt_cloud * gen_point_cloud(FILE * hndl_opendrive, _
                                  FILE * hndl_objdimensions, _
                                  struct Veh_positions pos);
```

openDRIVE file: geometry of the road, what object is located where

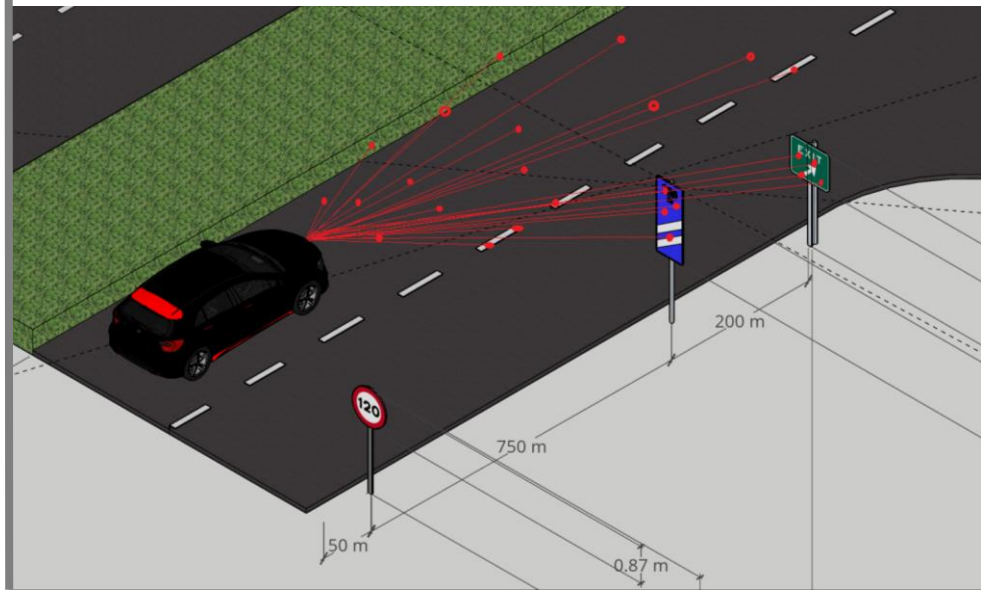


3D model (.skp file/alternative defined by you): physical dimensions of the objects



gen_point _cloud

*Structure containing coordinates of point cloud for traffic signs, signals, lane markings & road side barriers**



Q&A