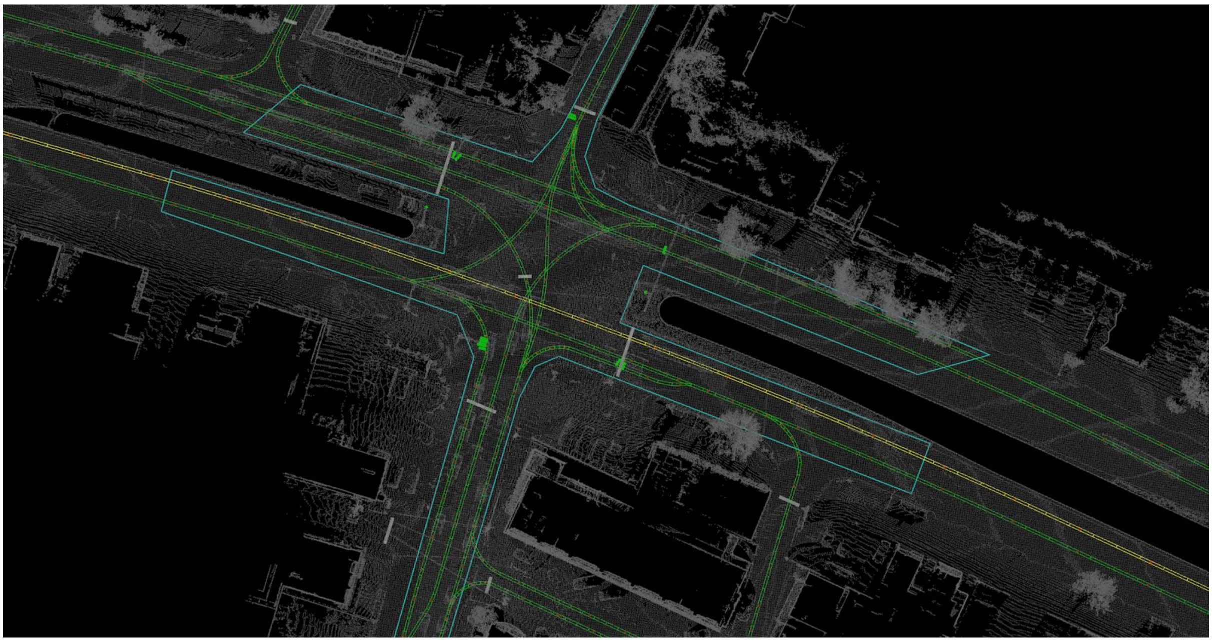
ASSURE Map Editor

Version 0.6

Update January 2021



Introduction

Assure map editor is an OpenGL and Linux based tool that enables creating and editing road network maps for self driving cars. The application supports map formats used for Autoware.AI. Autoware is an open source framework for autonomous cars.

Road network maps are an essential part of self driving cars technology. Some call it High Definition Maps HD Maps. Others call it Vector Maps.

Creating these maps is difficult and takes a lot of time. This application is developed to make creating and editing these types of maps as objective and as easy as possible.

Autoware.AI mainly uses AISIN technology maps, named “Vector Maps”. OpenPlanner is a complete integrated planner for self driving cars. It is part of Autoware.AI. It can use Vector Maps but because the format is commercial only provided maps are available. Since Autoware release 1.13 the open source map format “lanelet2” is introduced and utilized in several nodes including OpenPlanner nodes.

In addition to lanelet2 and to make it easy for developers and researchers of self-driving cars, OpenPlanner supports customized xml based map format with .kml extension.

Using ASSURE map editor, users can convert from vector map to kml format. Import one or more .csv files with path information then construct a kml map. Also using point cloud map to create kml map from scratch. Importing and exporting lanelet2 maps that work with Autoware.AI is now supported. Since version 0.6 a better support for OpenDRIVE map format import is introduced.

Supported Format

Imports:

1. **.csv format,** comma separated text based data files contains waypoint information (x,y,z,theta,v)
2. **.kml format**, customized xml based file. Designed and optimized to be used with OpenPlanner
3. **Vector Map**, collection of .csv files composing the map. This is a proprietary map format developed by AISAN Technology. TierIV provides an online editor for that format
4. **Lanelet2,** is an xml based file (.osm). It is open source map format, It is designed to utilize high-definition map data in order to efficiently handle the challenges posed to a vehicle in complex traffic scenarios. The modifications to the standards introduced by TierIV are also considered
5. **OpenDRIVE,** is an open format specification to describe a road network's logic (.xord). Its objective is to standardize the logical road description to facilitate the data exchange between different driving simulators
6. **.PCD files**: point cloud map files. To be used as a guideline for map creation

Exports:

1. OpenPlanner **.kml** format
2. Google Earth standard **.kml** format
3. Lanelet2 **.osm** format

Basic Functionality

The map editor is currently in beta version, only basic editing functionality are supported:

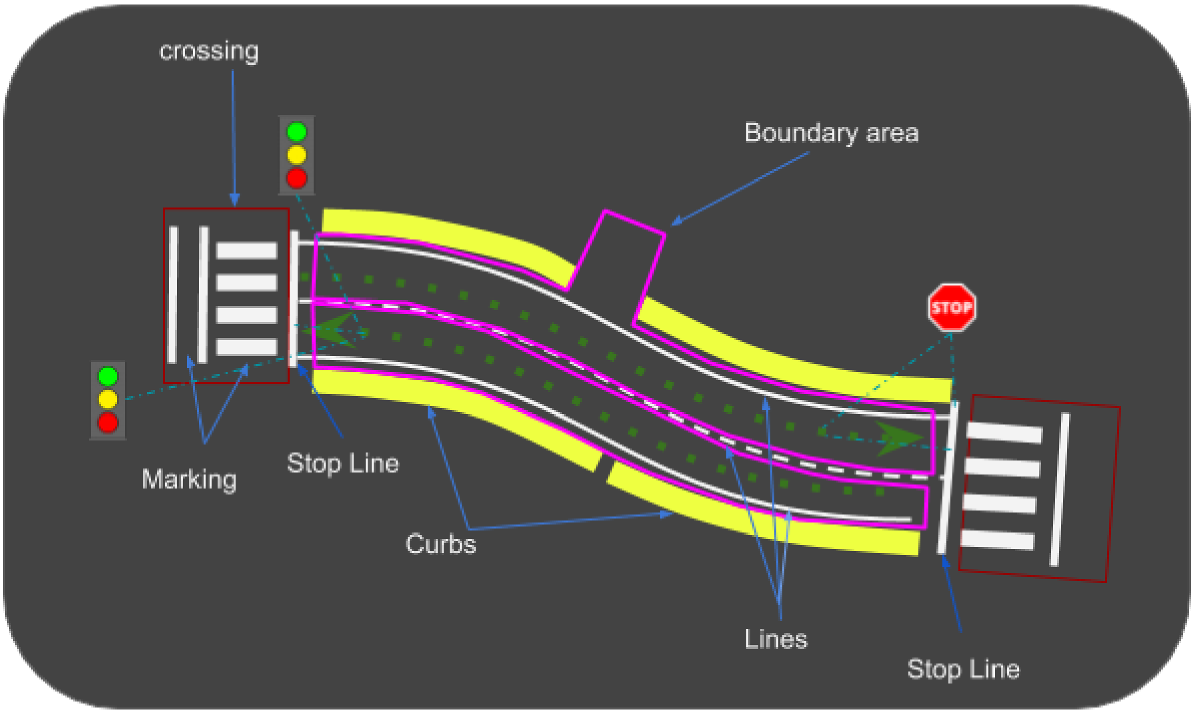
General Mapping Functions

1. Creating road network maps from scratch
2. Creating road network maps from .csv path file(s)
3. Creating road network maps guided by point cloud map
4. Importing and editing existing map formats such as (.kml, lanelet2 .osm, OpenDRIVE .xodr, Vector Map .csv)
5. Merge two .kml maps into one map
6. Converting from one path or collection of paths file(s) .csv format to .kml format.
7. Export map formats (Google Earth .kml, OpenPlanner .kml, Lanelet2 .osm)
8. Find parallel lanes and set lane change enable parameter

Supported Road Network Map Semantics

The internal road network map format consists of the following Items, an example is shown in the following figure.

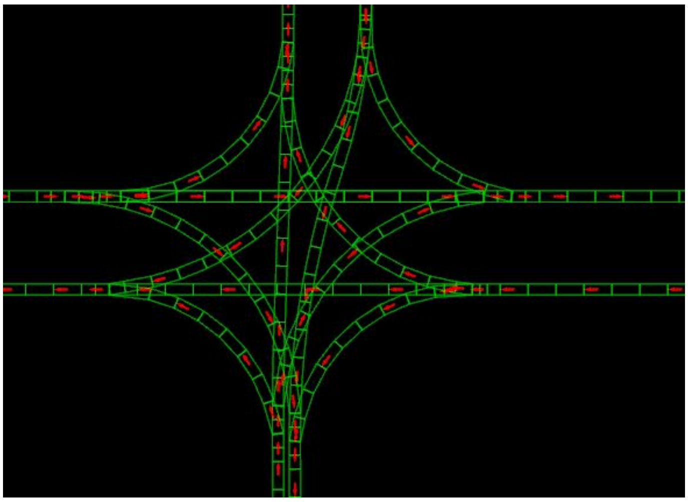
- Lane
- Line
- Stop Line
- Curb
- Boundary
- Crossing
- Traffic Sign
- Traffic Light



- ❏ Lane is the basic road network structure.
- ❏ Branching and merging only through the beginning and the end of the lane.
- ❏ Lanes consist of waypoints.
- ❏ Waypoints has multiple unit informations such as (velocity, pose, width) at each point.
- ❏ Waypoint connects the lane to the stopline.
- ❏ Waypoint is considered the node in the network graph.
- ❏ Waypoint has connection information such as (next points, previous points, left points, right points)
- ❏ Waypoints are lane center, Green dots.

- ❏ Boundary area (wayarea) border of defined space, Pink line.
- ❏ Curbs represent road and sometimes lane separation.
- ❏ Lines, a sequence of points representing painted lines such as white lines.
- ❏ Crossing is a boundary to a crosswalk area, red rectangle.
- ❏ Traffic lights can be (red, green yellow, left arrow, right arrow, forward arrow, crossing red, crossing green)

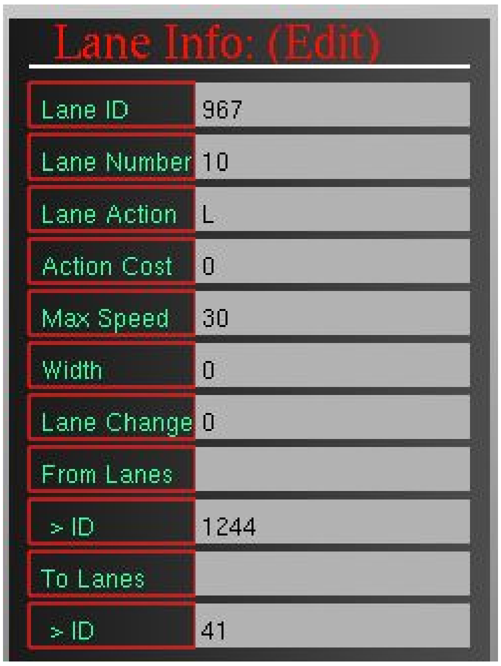
Lanes - Driving centerlines



Lane properties

Lanes consist of waypoints, at least 2 waypoints. The following table explains the lanes properties that could be set using the ASSURE map editor. The figures show how the lanes will look in the UI. The red arrows represent the lane direction.

Property	Description	Comments	Validation
LaneID	Unique Number Identifier		> 0
LaneNumber	Order from left to right in case of multiple parallel lanes	Not utilized	0,1,2 ..
Action	Semantic lane direction	Used in global planning	Left, Right, Forward
Action Cost	Cost of choosing certain action (actions and costs are pairs)	Used in global planning in addition to the distance cost	>= 0
Width	General width of the lane	If individual waypoints width are > 0 , this value will be ignored	
Speed	Maximum driving speed of the lane in k/h	If individual waypoints (v) > 0, this value will be ignored	
Type	Lane type	Not utilized in the planner yet, all lanes are driving lanes	
Lane change	If lane change is possible of not	Only as indication, left/right waypoints are used to calculate lane change.	1 if lane change is possible, 0 otherwise
From lanes	List of lanes ids that lead to this lane	Used for connecting the network graph	Empty, if this is first lane
To Lanes	List of lanes ids that branch from this lane	Used for connecting the network graph	Empty, if this is last lane



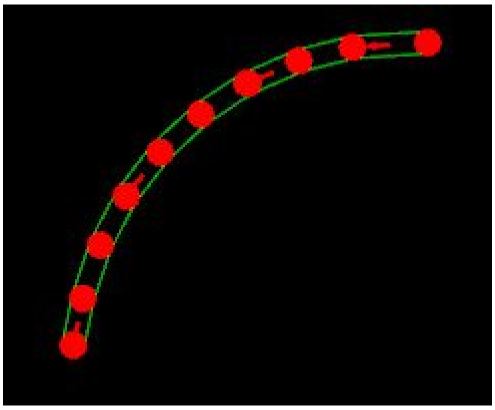
Lane Info: (Edit)

Lane ID	967
Lane Number	10
Lane Action	L
Action Cost	0
Max Speed	30
Width	0
Lane Change	0
From Lanes	
> ID	1244
To Lanes	
> ID	41

Lane supported function:

1. Insert new Lane.
2. Remove selected Lane
3. Edit existing Lane:
    a. Move lane
    b. Edit linking information for the lane
    c. Smooth the lane
    d. Split the lane into two lanes
    e. Merge lane with next lane to become one lane
    f. Flip waypoint, reverse lane direction
    g. Find the parallel lane on left and set it as left lane
    h. Find the parallel lane on right and set it as right lane
    i. Redistribute lane’s waypoint with specific resolution set in the Map Information dialog

Waypoints

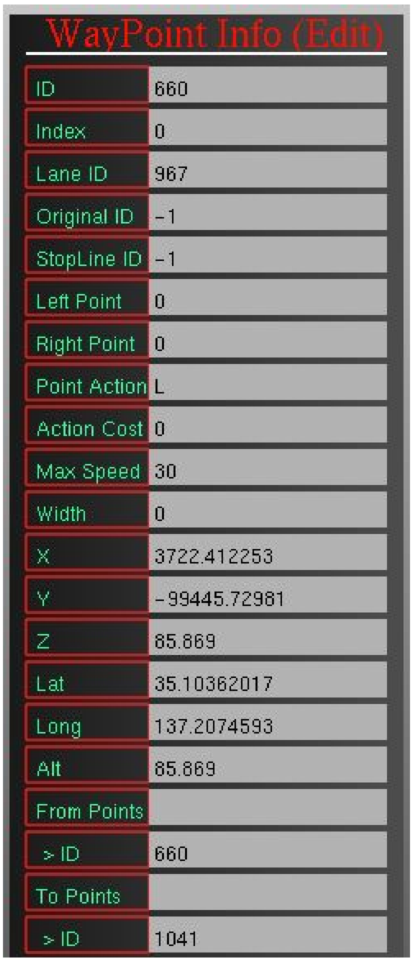


Waypoint Properties

Waypoints are the geometry representation nodes of the lanes centerlines.

Property	Description	Comments	Validation
ID	Unique Number Identifier		> 0
LaneId	Which lane this waypoint belongs to		> 0
StopLine Id	Id of the intersecting stop line	Each lane can have multiple stop lines, so their Ids are connected to the closest waypoints.	>= 0
Action, Cost	Same as in Lanes, but for individual waypoints	If this value is 0, planner will use the value from the lane	
Width	Width of the lane at this waypoint		
Max speed	Maximum driving speed of at this waypoint in k/h		
Left waypoint	Id of the closest waypoint in the left lane	Enable or disable lane change	0 if no left lane, or lane change is disabled, Otherwise > 0
Right waypoint	Id of the closest waypoint in the right lane	Enable or disable lane change	0 if no right lane, or lane change is disabled, Otherwise > 0
Pose	Geographical position of the point (x,y,z, longitude, latitude, altitude) + direction	lat , long , alt are used for coordinate projection only.	Lat, lon, alt are optional

From points	List of waypoints' ids that lead to this waypoint	Used for connecting the network graph	Empty, for first waypoint of first lane Multiple, if previous lanes exists Only 1, if in the middle of the lane
To points	List of waypoints' ids that branch from this waypoint	Used for connecting the network graph	Empty, for last waypoint of last lane Multiple, if next lanes exists Only 1, if in the middle of the lane



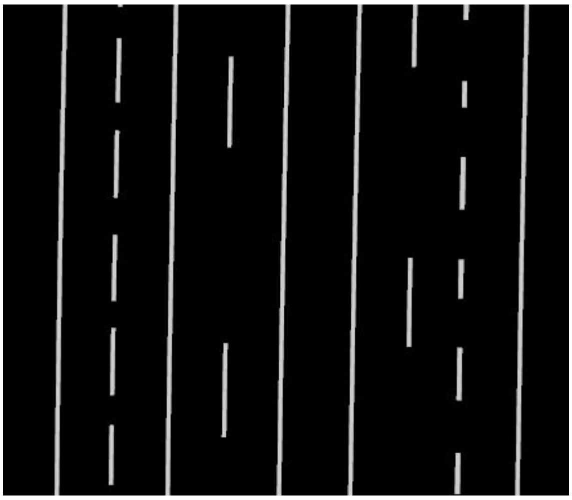
WayPoint Info (Edit)

ID	660
Index	0
Lane ID	967
Original ID	-1
StopLine ID	-1
Left Point	0
Right Point	0
Point Action	L
Action Cost	0
Max Speed	30
Width	0
X	3722.412253
Y	-99445.72981
Z	85.869
Lat	35.10362017
Long	137.2074593
Alt	85.869
From Points	
> ID	660
To Points	
> ID	1041

Waypoint supported functions:

1. Add waypoint
2. Remove waypoint
3. Move waypoint
4. Edit linking information for the waypoint

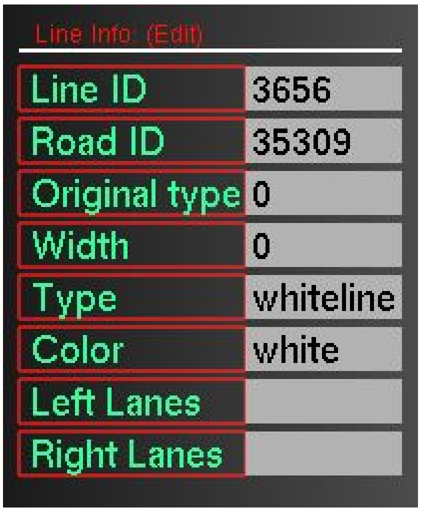
Lines - White and yellow lines



Line Properties

Road marking representing lanes and road edges

Property	Description	Comments	Validation
ID	Unique Number Identifier		> 0
RoadID	Related road		> 0
Width	Width of the line	White line must have one width	If 0, use the standard white line width of 15 cm
Type	White line alwayes	Other types will be represented as markings	
Color	White or Yellow	Only two line colors are supported	



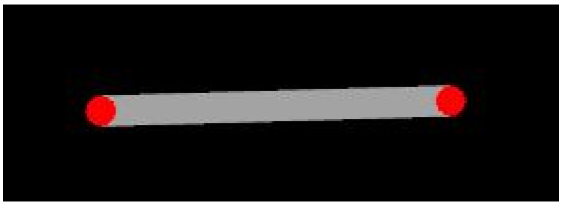
Line Info (Edit)

Line ID	3656
Road ID	35309
Original type	0
Width	0
Type	whiteline
Color	white
Left Lanes	
Right Lanes	

Lines Supported Functions:

1. Insert new line
2. Move line
3. Delete line
4. Smooth the line
5. Split the line

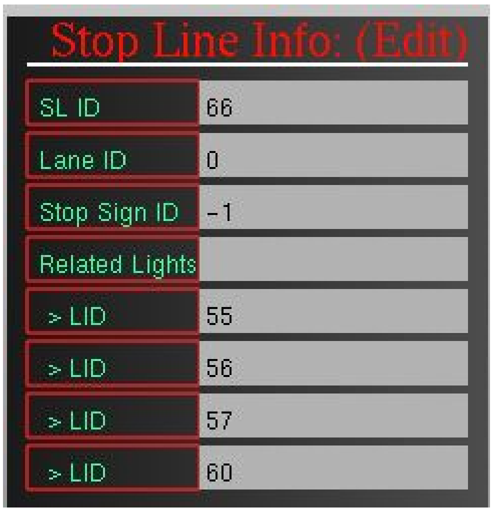
Stop Line



Stop Line properties:

Stopping point for traffic control. It consists of 2 or more points.

Property	Description	Comments	Validation
ID	Unique Number Identifier		> 0
LaneID	Related lane that this stop line controls stopping at	If not assigned to lane, it will never be used	> 0
Stop Sign ID	ID if related stop sign	Can't be used with traffic lights. Either traffic light or stop sign is used	-1 if stop line used for traffic light, otherwise > 0
Traffic Lights Ids	List of lights' ids that control this stop line	Connect multiple traffic light to stop line	Empty if stop line is used for stop sign, otherwise include one or more ids



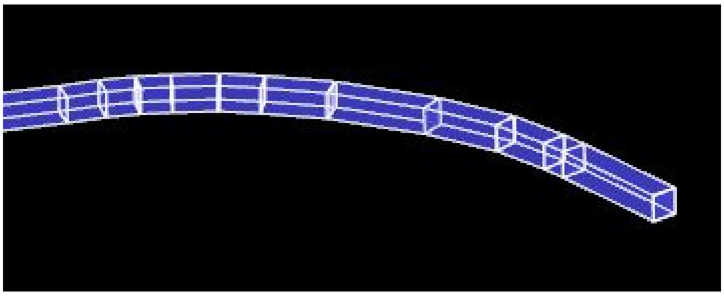
Stop Line Info: (Edit)

SL ID	66
Lane ID	0
Stop Sign ID	-1
Related Lights	
> LID	55
> LID	56
> LID	57
> LID	60

Stop Line supported functions:

1. Insert new Stop Line
2. Edit existing Stop Line:
    a. Move line end points
    b. Edit linking information
3. Remove selected Stop Line

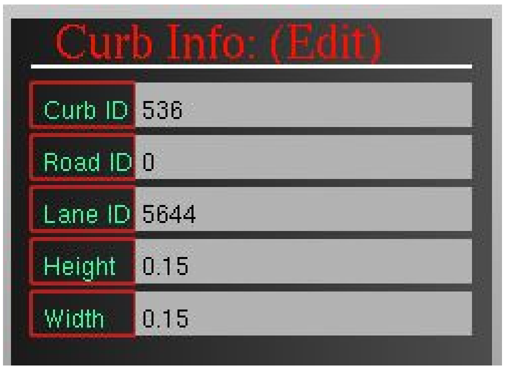
Curb



Curb Properties:

High undrivable edge of the road. (usually +15cm), used to assist the difficulty of sensing low objects, and curbs that are very close to the vehicle. The curb section is defined by a list of points.

Property	Description	Comments	Validation
ID	Unique Number Identifier		> 0
LaneID	Closest lane to the curb		0
Height	Curb Height in meters		
Width	Curb width in meters		



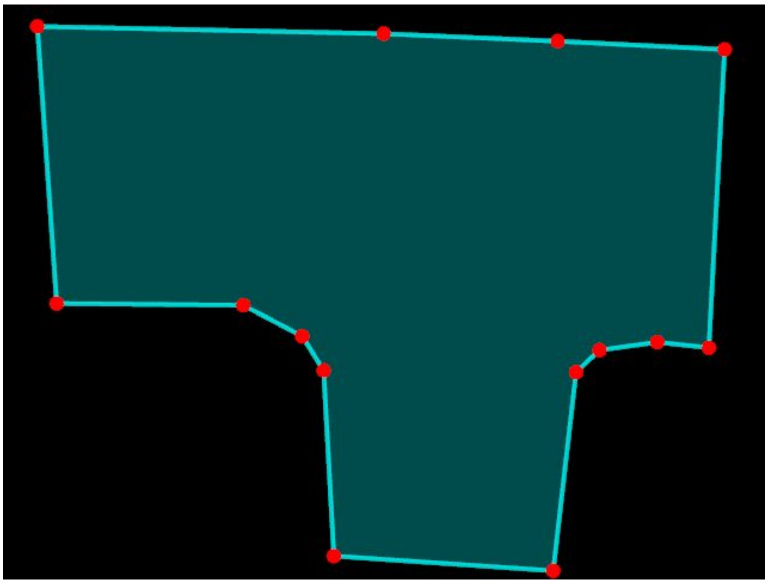
Curb Info: (Edit)

Curb ID	536
Road ID	0
Lane ID	5644
Height	0.15
Width	0.15

Curb Supported functions:

1. Insert new Curb
2. Edit existing Curb
3. Remove selected Stop Line

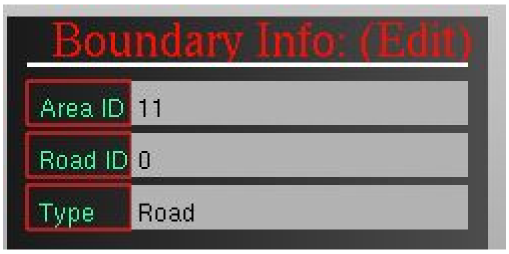
Boundary



Boundary Properties:

Area in the map is defined by a list of points forming a polygon, it is used in object filtering, defining parking places, pedestrian sidewalk and so on.

Property	Description	Comments	Validation
ID	Unique Number Identifier		> 0
Type	Boundary type	To determine the functionality to be used with, for example road boundary use to filter objects on the road, but sidewalk boundary is used to filter pedestrians and bicycles on the sidewalk. Parking area boundary used for path planning to park or leave parking spots.	
Center	A point represents the center of the area	Use to quickly search for closest boundaries around the vehicle	



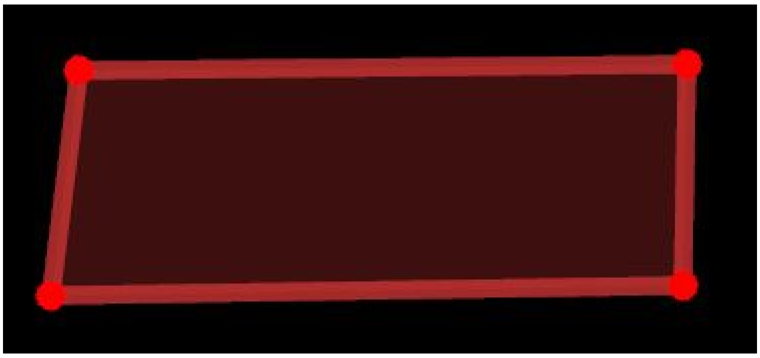
Boundary Info: (Edit)

Area ID	11
Road ID	0
Type	Road

Boundary supported functions:

1. Insert new Boundary
2. Edit existing Boundary
3. Remove selected Boundary

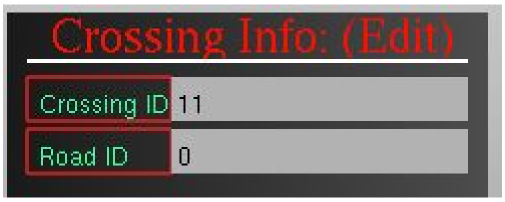
Crossing



Crossing Properties:

An area of just 4 points. Used to identify a pedestrian crossing point.

Property	Description	Comments	Validation
ID	Unique Number Identifier		> 0



Crossing Info: (Edit)

Crossing ID	11
Road ID	0

Crossing supported functions:

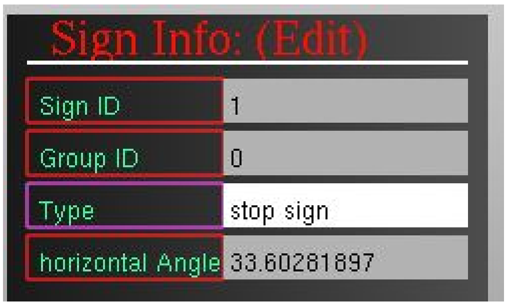
1. Insert new Crossing
2. Edit existing Crossing
3. Remove selected Crossing

Traffic Sign

TrafficSign Properties:

Traffic sign is used to specify the position and type of a Road sign. Currently only Stop Sign is utilized.

Property	Description	Comments	Validation
ID	Unique Number Identifier		> 0
Type	Type of the sign		
Horizontal Angle	Absolute orientation angle for the sign, in degrees		
Vertical Angle	Tilt angle of the sign, assumed 0 all time.		



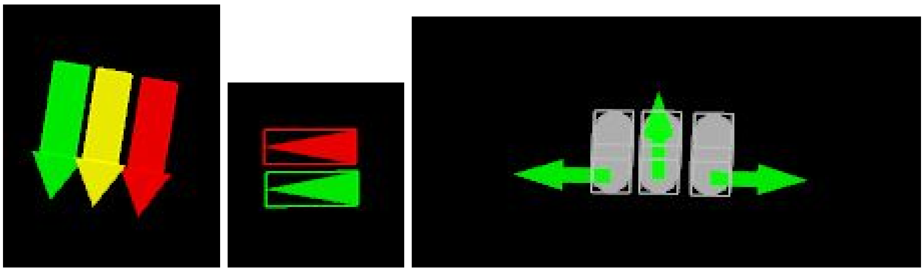
Sign Info: (Edit)

Sign ID	1
Group ID	0
Type	stop sign
horizontal Angle	33.60281897

TrafficSign supported functions:

1. Insert new sign
2. Edit existing sign
    a. Select sign type
    b. Rotate sign to the correct angle
    c. Move sign
3. Remove selected sign

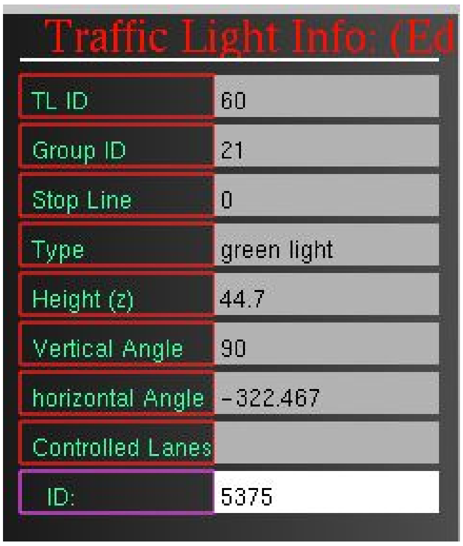
Traffic Light



TrafficLight Properties:

Traffic sign is used to specify the position and type of a Traffic light bulb. It can affect multiple lanes. Types are [green, yellow, red, left, forward, right] for cars and [green, red] for pedestrians.

Property	Description	Comments	Validation
ID	Unique Number Identifier		> 0
Type	Type of the light bulb		
Stop Line	Id of the controlled stop line		
Horizontal Angle	Absolute orientation angle for the light In degrees		
Vertical Angle	Absolute Tilt angle of the light In degrees		
Height	Height of the light bulb center from the road surface. in meters		
Lane Ids	List of lane Ids that are controlled by this traffic light		At least one lane.



Traffic Light Info: (Ed

TL ID	60
Group ID	21
Stop Line	0
Type	green light
Height (z)	44.7
Vertical Angle	90
horizontal Angle	-322.467
Controlled Lanes	
ID:	5375

TrafficLight supported functions:

1. Inset Traffic Light
2. Edit existing traffic light:
    a. Move traffic light center point
    b. Edit linking information
3. Remove selected Traffic Light

System Installation

Native Installation:

Download the application folder and binaries from the GitHub repository. But to be able to run the application it is important to install the prerequisites first. The application is tested on Linux 18.04. For more information please go to the GitHub page.

Prerequisites

- OpenGL
- Tinyxml
- PCL 1.7

Steps:

1. Install OpenCV 2.4 don't forget to checkout ver 2.4.13.7, default git clone download ver 4.0
2. Install ROS melodic
3. Install libtinyxml, freeglut3, libglew, libpcl1, libpugixml-dev, libgeographic, libplib, libglm
4. Install ros-melodic-pcl-ros

Library link issue: export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:libs
If still there are library linking problems use the Fast fix

Docker Installation:

Another way of installation is to use Docker:

1. Install docker to your system
2. $ sh build-docker.sh (will build the image on your system, the current GitHub branch name will be the docker image name)
3. $ .run-docker (will run the assure mapping tools container with GUI support)
   a. You can use parameters with run-docker (-b, -w)
   b. -b followed by the branch name, use this when branch is not “master”
   c. -w followed by a local machine full directory path name, use this to set a working directory so you can load and save maps.

Run the application

From starting with GUI, Map Editing mode:

```
$  ./assure_map_editor
```

OpenDRIVE conversion mode:

For only converting from OpenDRIVE to other formats use command line parameters as follows:

Mandatory parameters:

1. Target Map type: -kml, -earth, -lanelet
2. Source map folder or file name, .xodr OpenDRIVE map file
3. Destination .kml or .osm map file name
4. Optional parameters, use - res followed by a distance in meter to specify the waypoint resolution in the converted lane.

Example:

```
./assure_map_editor -lanelet city01.xodr city01.osm -res 0.5
```

GUI Usage Manual

The main UI is divided into two sections, the left wide section which shows the map and enables navigation and selection.The right section which shows detailed information about hovered and selected map items.

Mouse Functions:

For the Map View:

1. Zoom In/out: mouse scrollwheel
2. Hover: display object information temporarily by point motion
3. Move left/right/top/down: mouse middle button pressed and mouse move
4. Select Map area, select one specific sub map rectangle by left click.
5. Select Object: left mouse click
6. Move waypoint: First click to select by left button, then left button down and move.

For The Info View:

1. Select Information Item: by left click.

Keyboard Functions:

For the Map View:

1. Delete a selected object (non waypoint object) . [ctrl+d+d]
2. Delete a selected waypoint. [-]
3. Add waypoint, when a waypoint is selected, it adds another waypoint in the mid distance to the next waypoint/ [+]
4. Smooth lane, when a lane is selected, apply one smooth step. [*]
5. Move point, when a point is selected. Arrows can move point to all directions. [top, bottom, left, right]
6. Exit sub map selection, when focus on the Map View area. [Esc]
7. Exit the whole map editor without saving, when focus on the outer border of the Map View area. [Esc]

For The Info View:

1. Move from information item to another by using the arrows. [top, bottom]
2. Delete information value. [Backspace]
3. Add information value. [numbers, characters]
4. Add information list item. Select list header then press. [+]
5. Remove the information list item. Select list item then press. [-]

6. Save changes to changed information, press [ctrl+s]. No automatic save to the map.