SENSR I 1.2

User Manual

Contents

1. Introduction

SENSR is a 3D perception software platform compatible with major 3D LiDAR sensors. It is the first industrial-grade software solution that provides scalable 3D perception for companies that seek to efficiently commercialize across industries and applications, and with the flexibility to choose LiDAR sensors and integrate multiple different sensors with ease. SENSR detects, classifies, tracks and predicts objects and segments ground, object and environment points out of raw 3D point cloud from sensors.

SENSR platform has 3 sub-platforms:

- SENSR I for human tracking application
- SENSR S for outdoor smart monitoring application
- SENSR M for automotive/robotics application.

2. System Setup

System Requirement

1. OS: Ubuntu 18.04 (Clean machine is recommended)

System Setup

1. Open Terminal in SENSR folder
2. Run **$ ./install.sh**
3. Reboot your computer

Sample Data

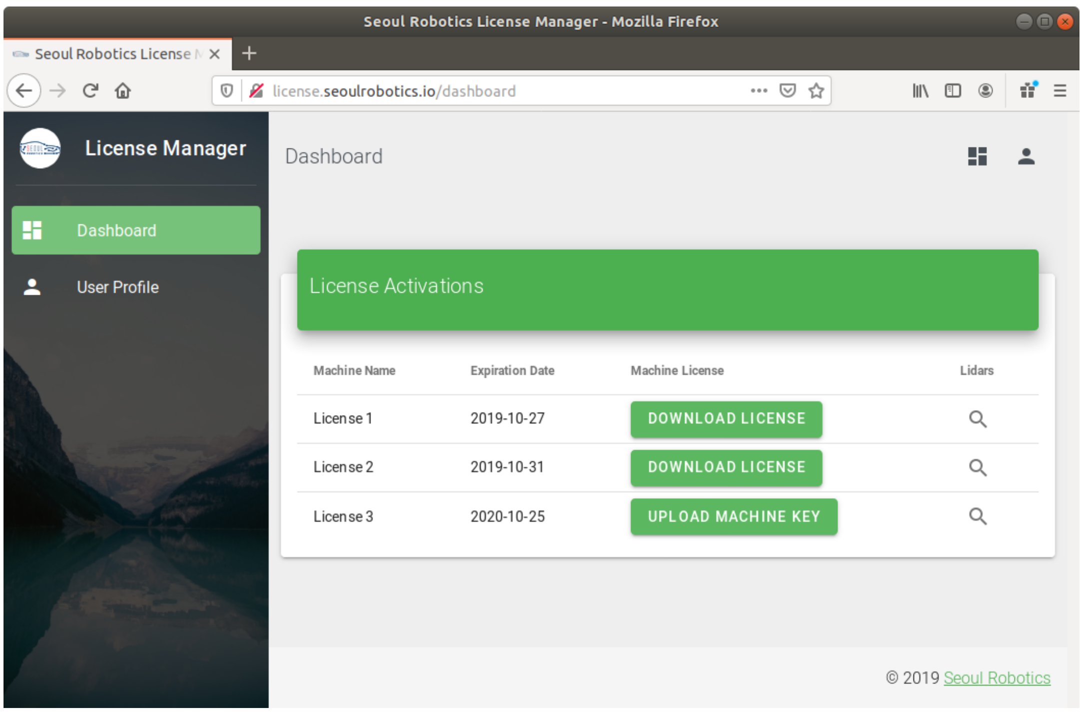
When you run install.sh script, sample data will be downloaded in the **samples** folder in the SENSR folder

1. sample_data.bag: sample input data to simulate Lidar point cloud input (see Section 4)
2. sample_map.png: sample map image for map loading function (see Section 5, Step 6)
3. sample_replay.re: sample replay file which you can load in Replay editor (see Section 7)
4. sample_output: sample output files (protobuf) from the sample rosbag data (see Section 10)

3. License Activation

1. Find generated **license_key.json** file in SENSR folder
2. Go to Seoul Robotics License Portal here: http://license.seoulrobotics.io/
3. Sign in to your account

   * **NOTE**: You have to set a new password when you sign-in for the first time.
4. Go to **Dashboard** and check available license



5. Click **UPLOAD MACHINE KEY** button of the license you want to activate and upload **license_key.json** file that was generated during installation.

   * **NOTE**: You can register only one machine to your license
6. When the **UPLOAD MACHINE KEY** button changes to **DOWNLOAD LICENSE**, click the button and download your license file (.lic) to your SENSR folder

   * **NOTE**: There is a known issue of upload machine key failure with some Firefox versions. If your **UPLOAD MACHINE KEY** doesn't work, use other browsers

such as Google Chrome or update your Firefox (run **$ sudo apt update && sudo apt install firefox** in your terminal).

4. Launch

Launch SENSR

1. Open Terminal in SENSR folder
2. Run **$ ./SENSR.sh**

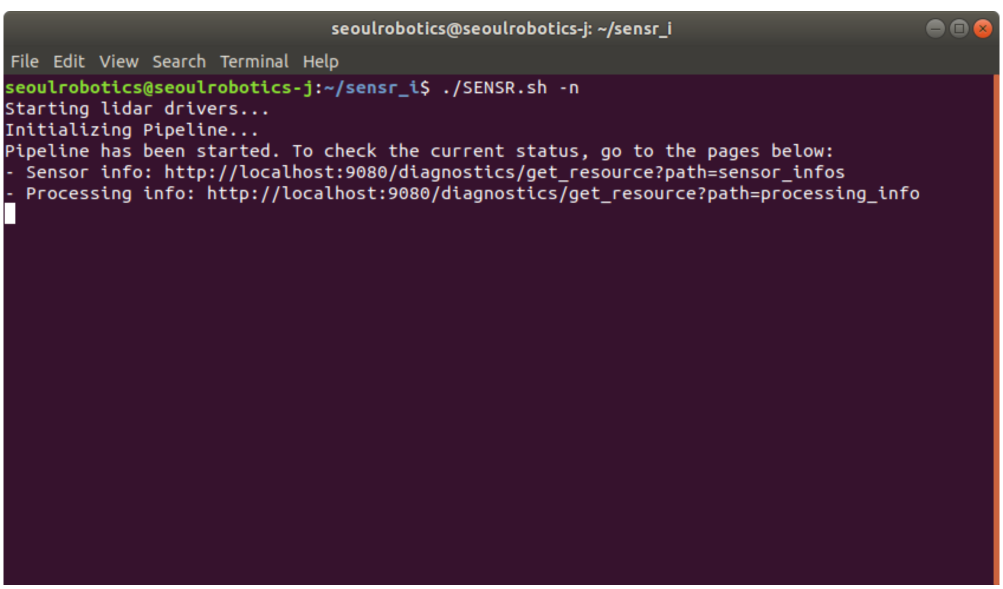
Tutorial: [Tutorial Video with Sample Rosbag Data]()

(Sensor Setup → Lidar Calibration → Zone Setup → Runtime → Replay Editor)

Launch SENSR with No-GUI Mode

If you have finished setting up sensors, calibration, and parameters and no longer need to see the 3D View, you can launch SENSR with no-GUI mode. SENSR will load your settings and process the Lidar data. You can check the sensor and processing status with the command line interface and web interface.

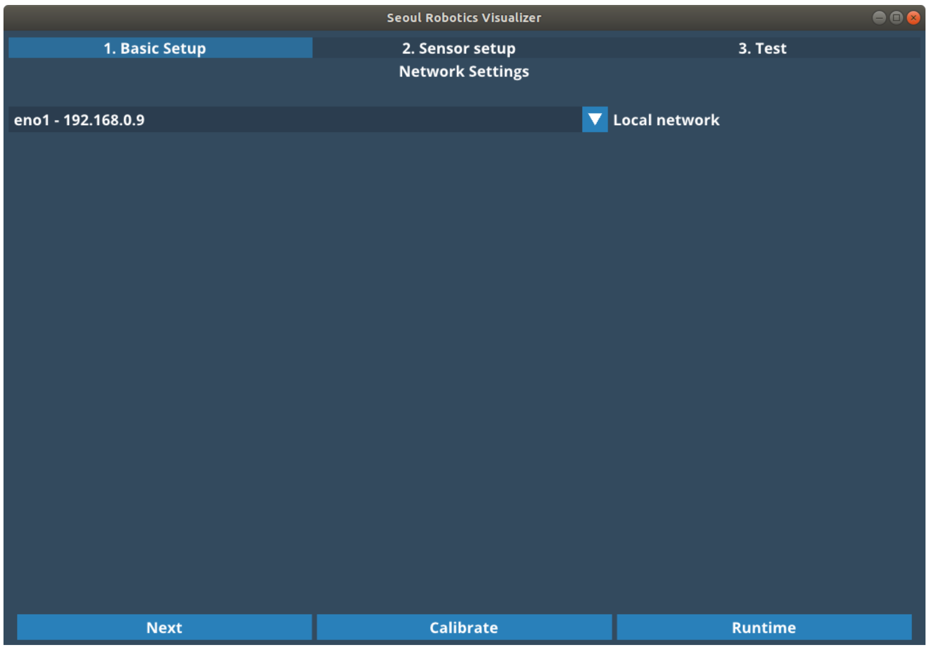
1. Open Terminal in SENSR folder
2. Run **$ ./SENSR.sh -n**



5. Sensor Setup

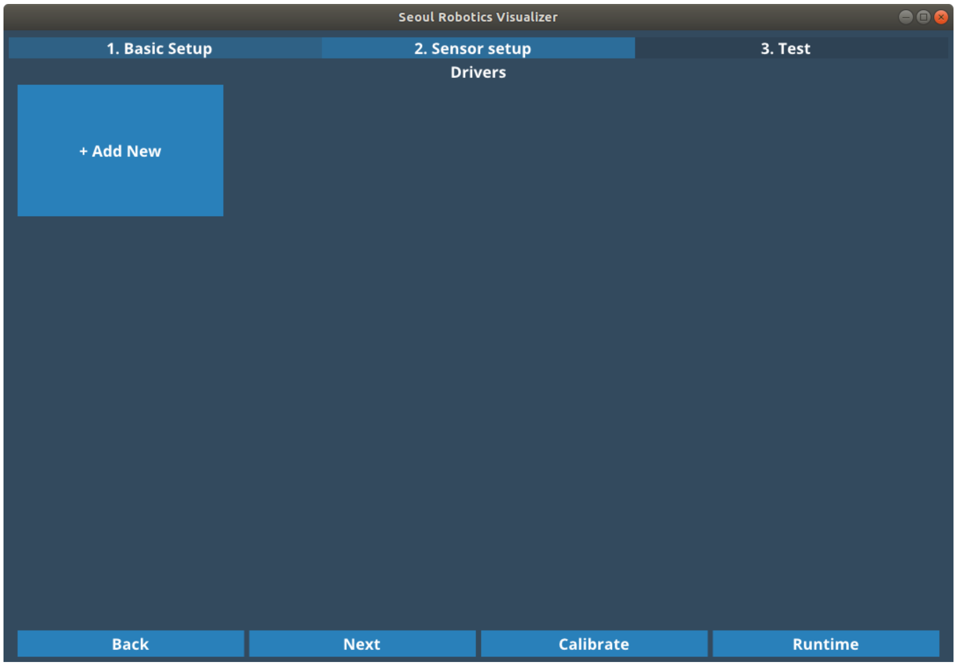
Step 1 : Basic Setup

1. Select your Host Machine IP address which is connected to the **same subnet** of Lidar sensors.

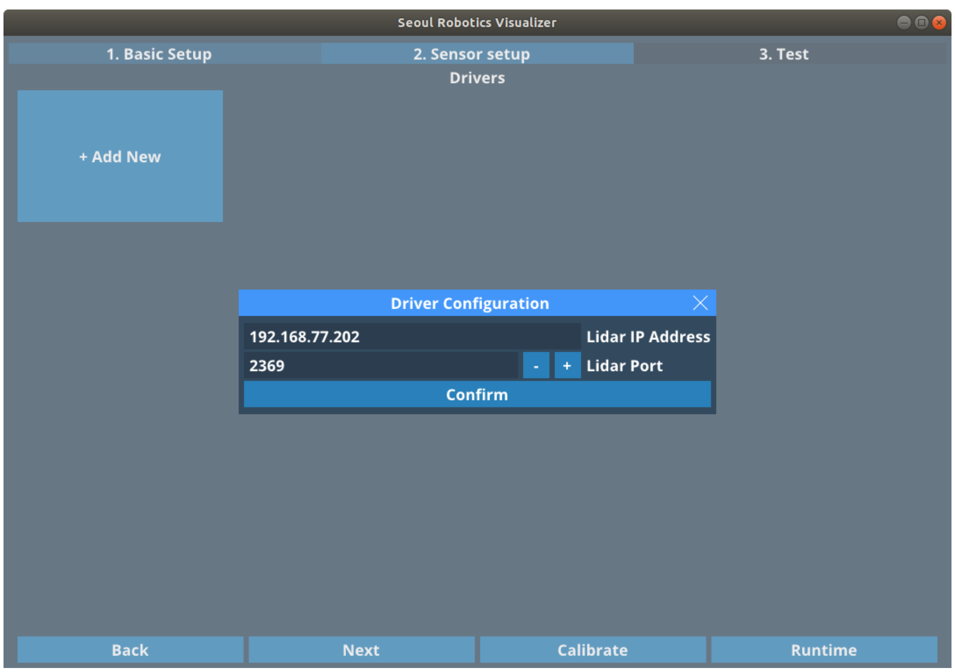


Step 2 : Sensor Setup

1. Click + **Add New** button to add a new sensor or rosbag.



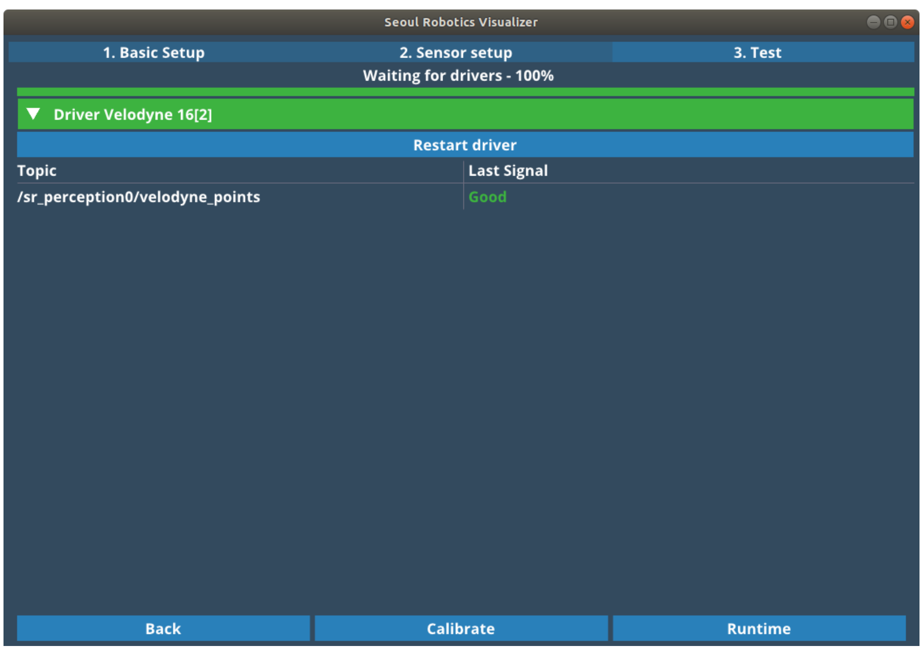
2. Type Lidar sensor's IP address and port number. In the case of rosbag, provide the path of the input rosbag file.



Step 3 : Test

1. Wait 3 - 5 seconds until sensors are connected.

2. Once all sensors are connected, click **Calibrate** to calibrate sensors or click **Runtime** to skip calibration.



Step 4 : Calibrate

Calibrate Lidar sensors in 3D space with 3D manual editor and tools in the **Tool** menu. You can go back to the **Sensor Setup** mode or go to the **Runtime** mode in the **Mode** menu.

1. Introduction

The perception result of the SENSR is based on its coordinate system represented by the axes in 3D View.

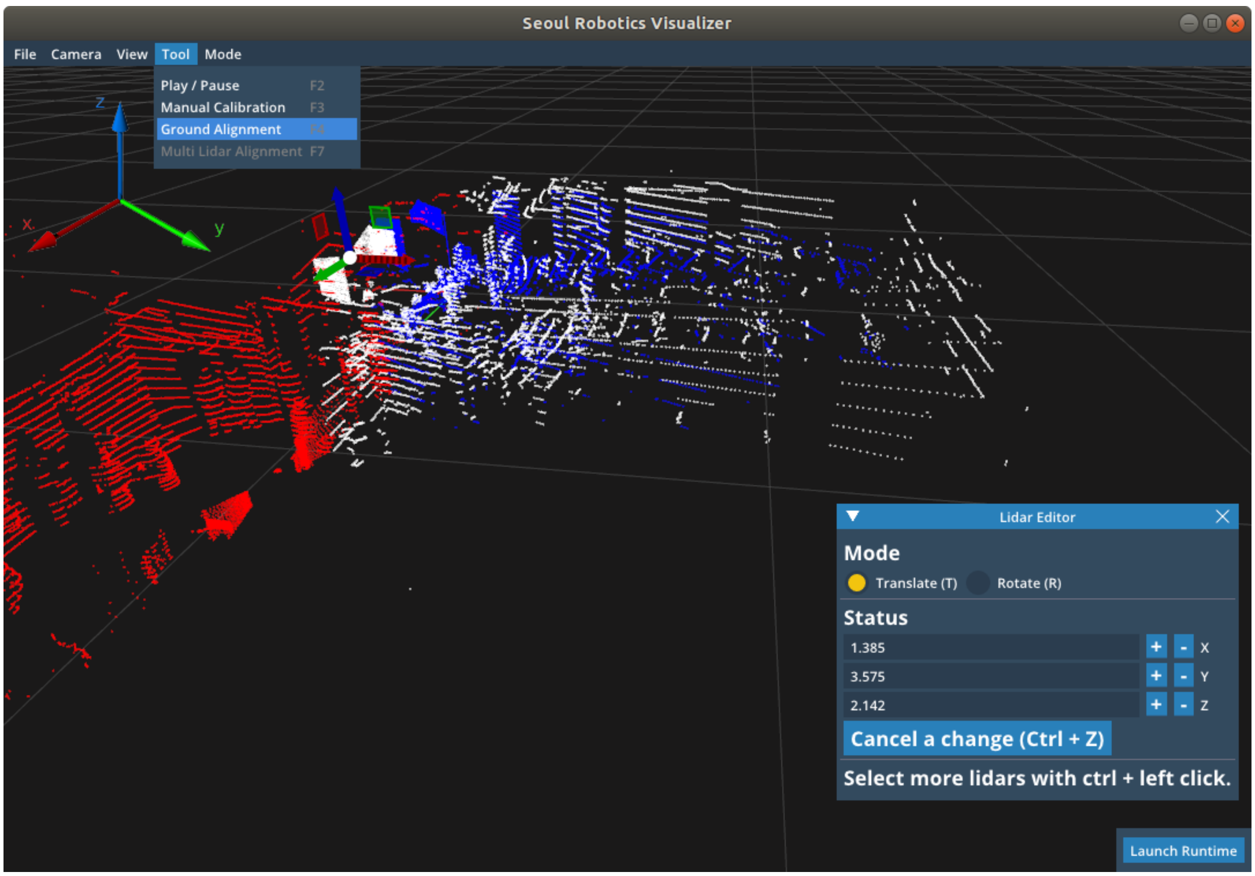
Here are the goals you need to achieve in the calibration step:
(1) Align the ground plane of the real world with SENSR's XY plane.
(2) Place the Lidar(s) in the right position and orientation in SENSR's coordinate system to align the perception result with your coordinate system.

With the 3D calibration anchor, text box and semi-auto calibration tools, you can easily rotate, translate and align the Lidar(s) as you want.

For more detailed step by step instruction of calibration step, please check **tutorial videos** for each step.

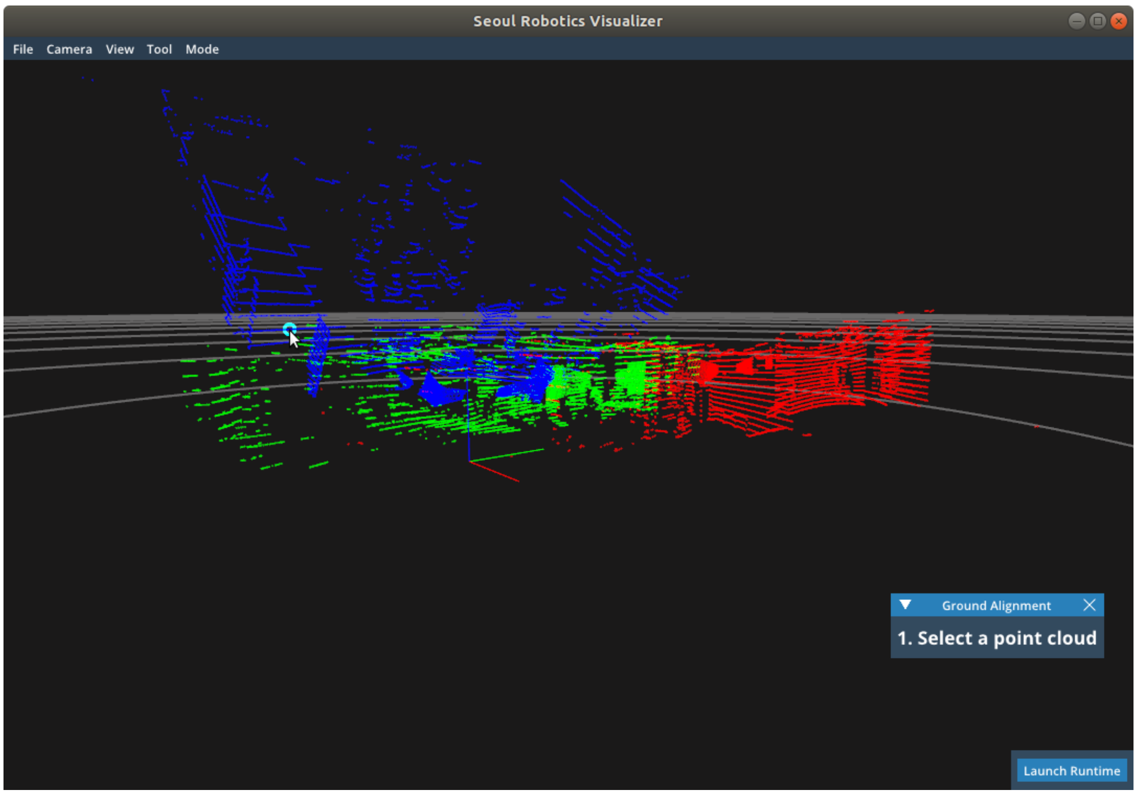
2. Ground Alignment ([Tutorial Video]())

The **Ground Alignment** tool belongs to the semi-automatic calibration tool family. The idea is to work with you for calibrating your Lidars through a step by step process to finally perform adjustments on our side. To begin with this tool, open the **Tool** menu at the top of the window and select the **Ground Alignment** option (or simply press **F4**).



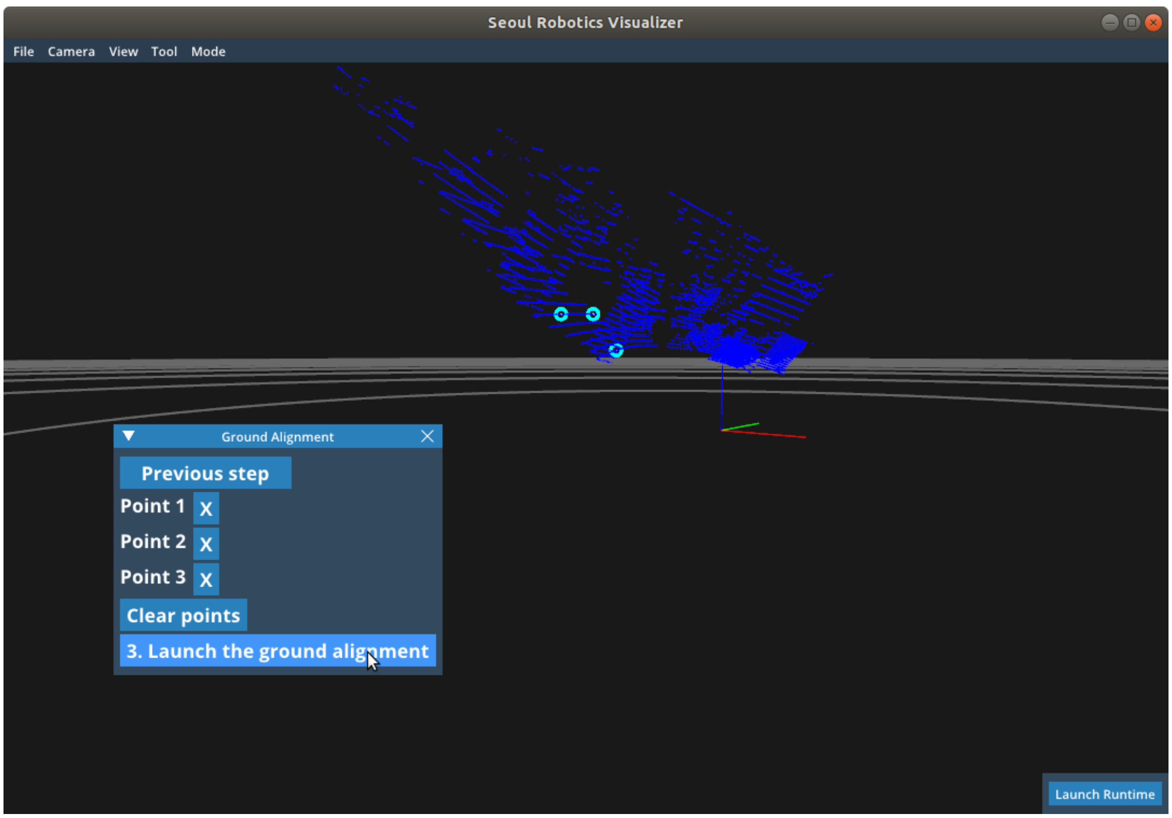
2.1. Point cloud selection

Select a Lidar you want to calibrate by clicking on its point cloud. If you have multiple Lidars in the scene, the non selected ones will disappear for the sake of an easier calibration.



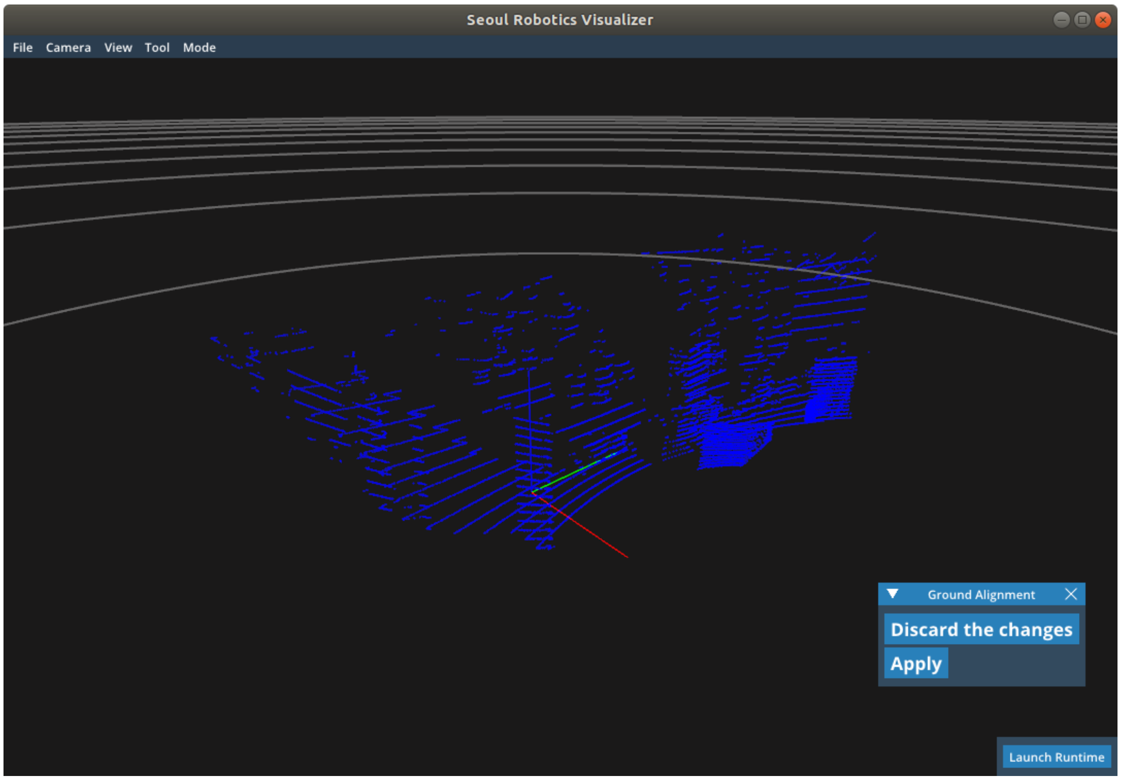
2.2. Point selection

Now we need you to select 3 points that should normally be on the ground. You can simply hover it with your mouse and then click it to select a point. You can always cancel a selection by re-selecting the point or by using the **Ground Alignment** window shown below. Once you have your 3 points, press **Launch the ground alignment** in the window to move to the next step.



2.3. Ground alignment

Now, this is our time to do the work. If we worked hard enough and the points selected were good, you should see your lidar with better alignment with the ground such as this picture.



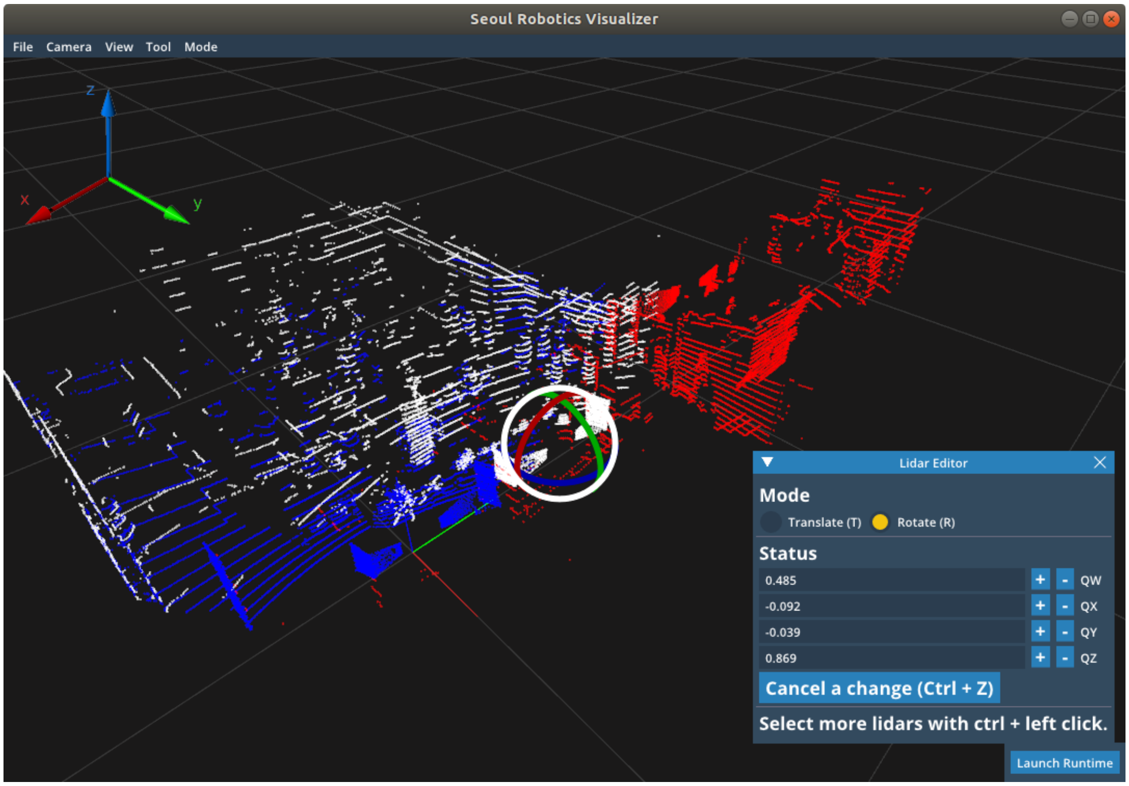
However, if it's not quite accurate you can always :

- Discard the changes and try from a fresh start or
- Apply the changes and do a new ground alignment on top of it.

The second option will be preferable most of the time.

3. Manual calibration ([Tutorial Video]())

The manual calibration tool allows you to move and adjust your Lidars with a 3D anchor (named gizmo) but also by editing the values themselves.



The gizmo allows you to rotate or translate a given Lidar. To do so, simply click on the point cloud related to the Lidar. When the point cloud becomes white a gizmo should appear. You can start clicking on its axis to move the lidar accordingly. If you want to be more precise with the positions you can also edit the Lidar position and orientation by tweaking the values in the **Lidar Editor** window. The change you make is always revertible with **Cancel a change** button or **Ctrl + z** combo.

Note that you can also edit multiple Lidars at the same time by pressing **Ctrl + Left button click** on additional point cloud. You can only translate the Lidars when you move multiple Lidars at the same time.

Step 5: Zone Setup (Optional)

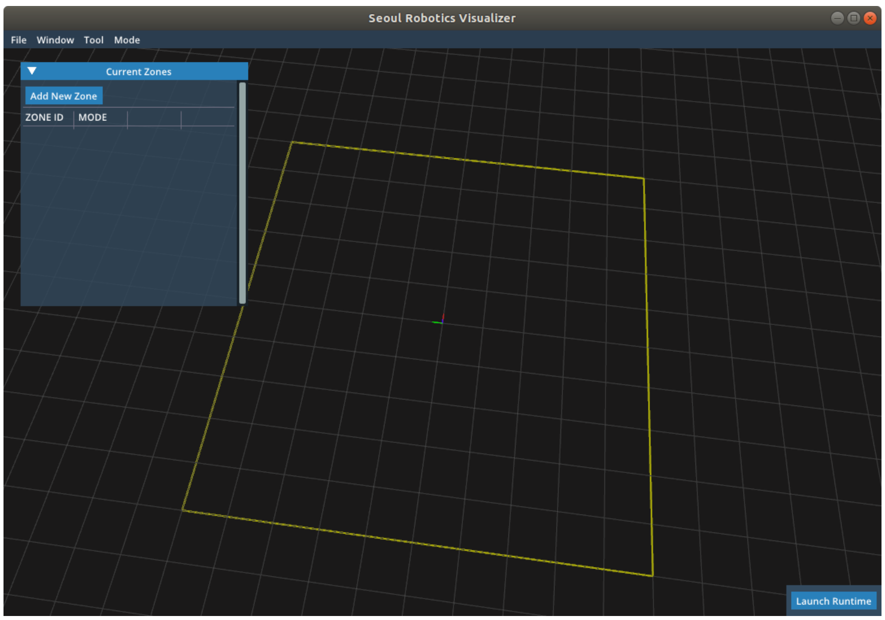
[Tutorial Video]

To make an event when an object is detected or to ignore objects in specific regions, you can set up zones in **Mode > Zone Setup**.

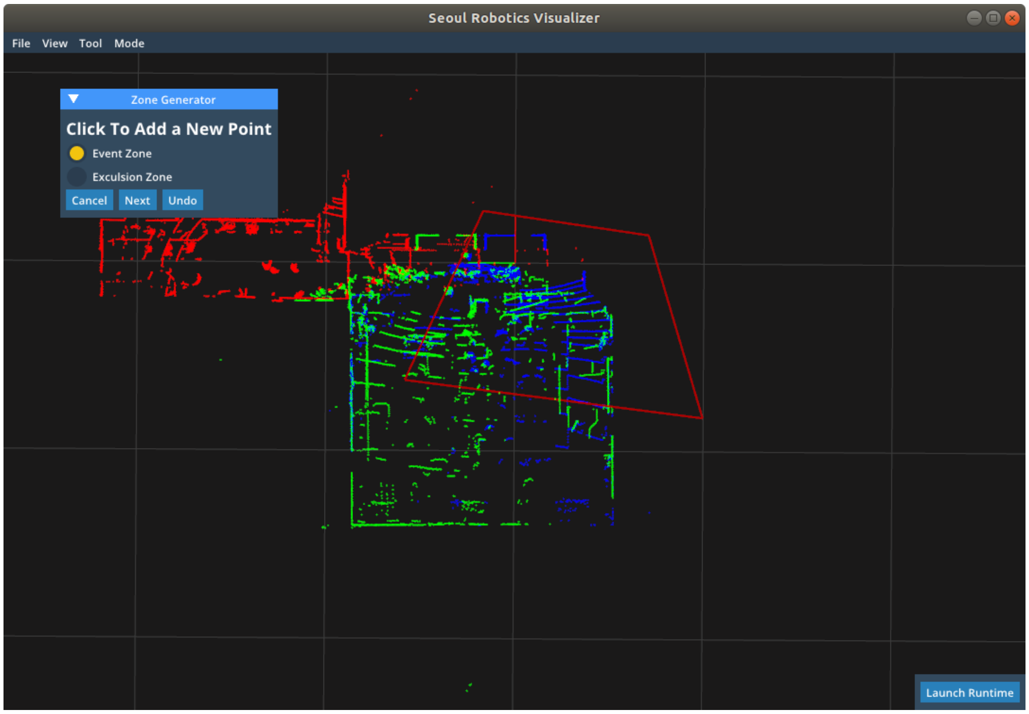
There are two types of zones: **Event Zone** and **Exclusion Zone**.

An **event zone** is a zone of interest. If an object is detected in an event zone, the object's ID will be associated with zone ID in SENSR's output.

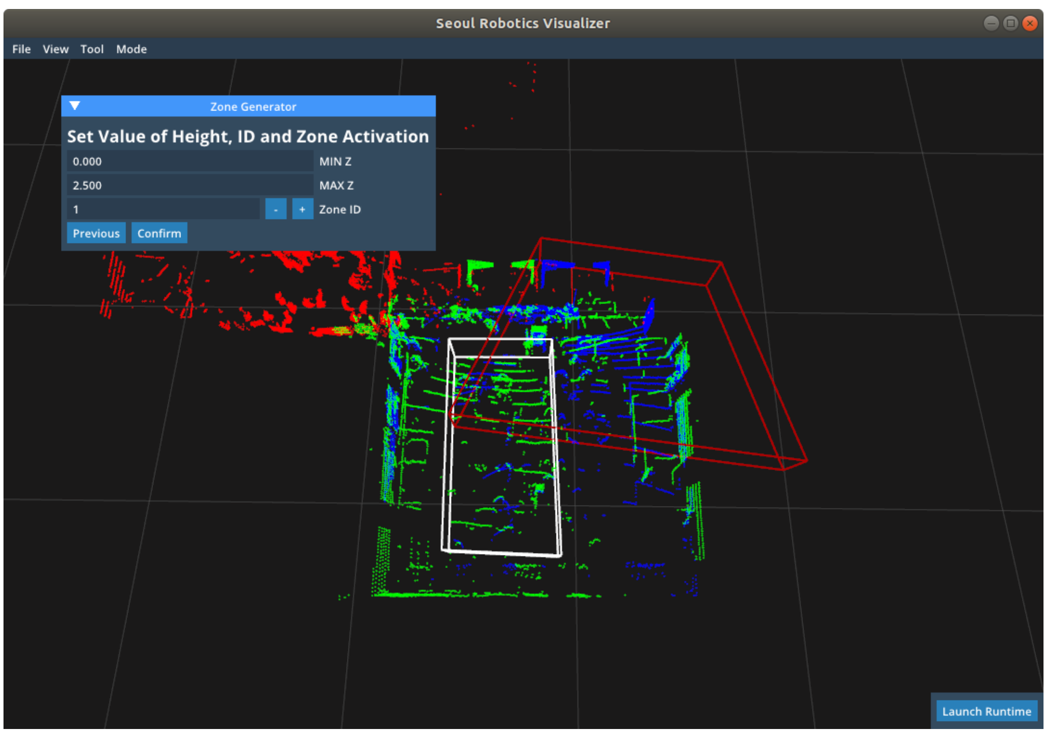
An **exclusion zone** is a zone you want to ignore. All objects detected in the exclusion zone are ignored by SENSR.



When you add a zone, you can draw a free shaped polygon in **a dark yellow rectangle** which is a boundary of SENSR detection range and set whether the zone is an event zone or exclusion zone. Note that two event zones cannot intersect with each other, whereas exclusion zones can intersect with all other zones without problems. In the case where an exclusion zone has an intersection with an even zone the exclusion zone has priority.



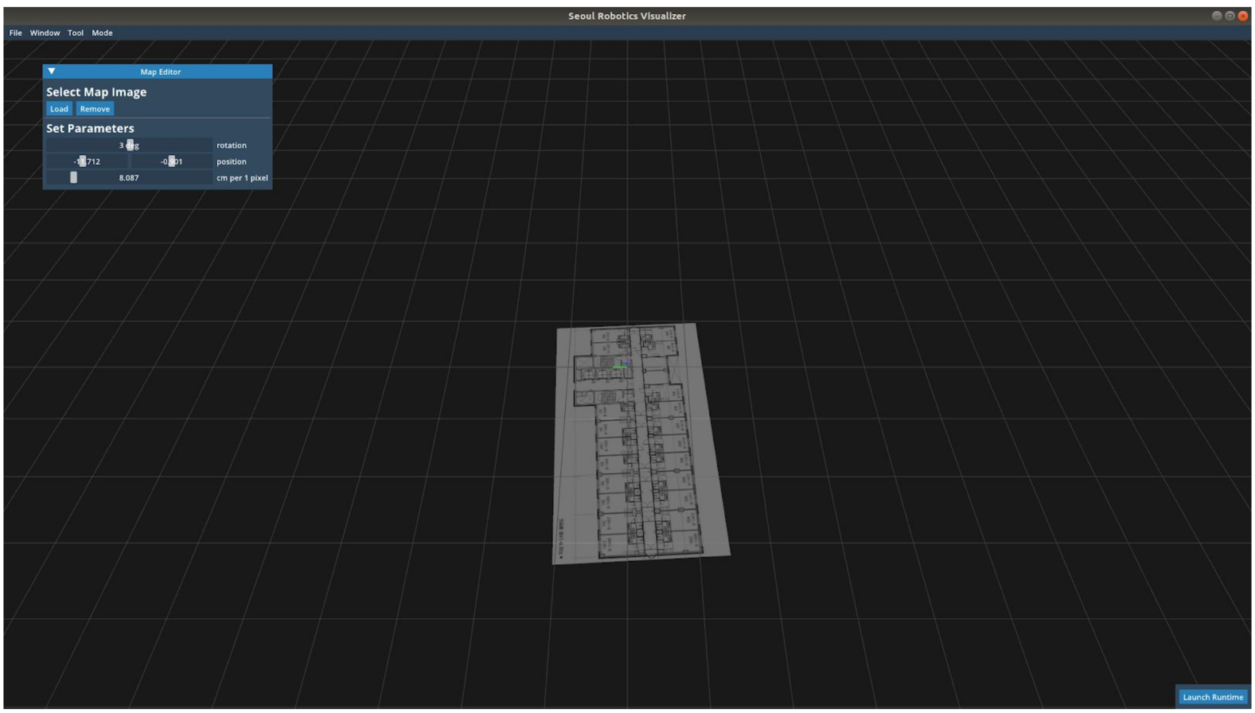
In the next step, you can set the height (z-direction) information and change the zone ID. Zone ID is not a unique ID and you can set anything you want.



Step 6: Map Setup (Optional)

[Tutorial Video]

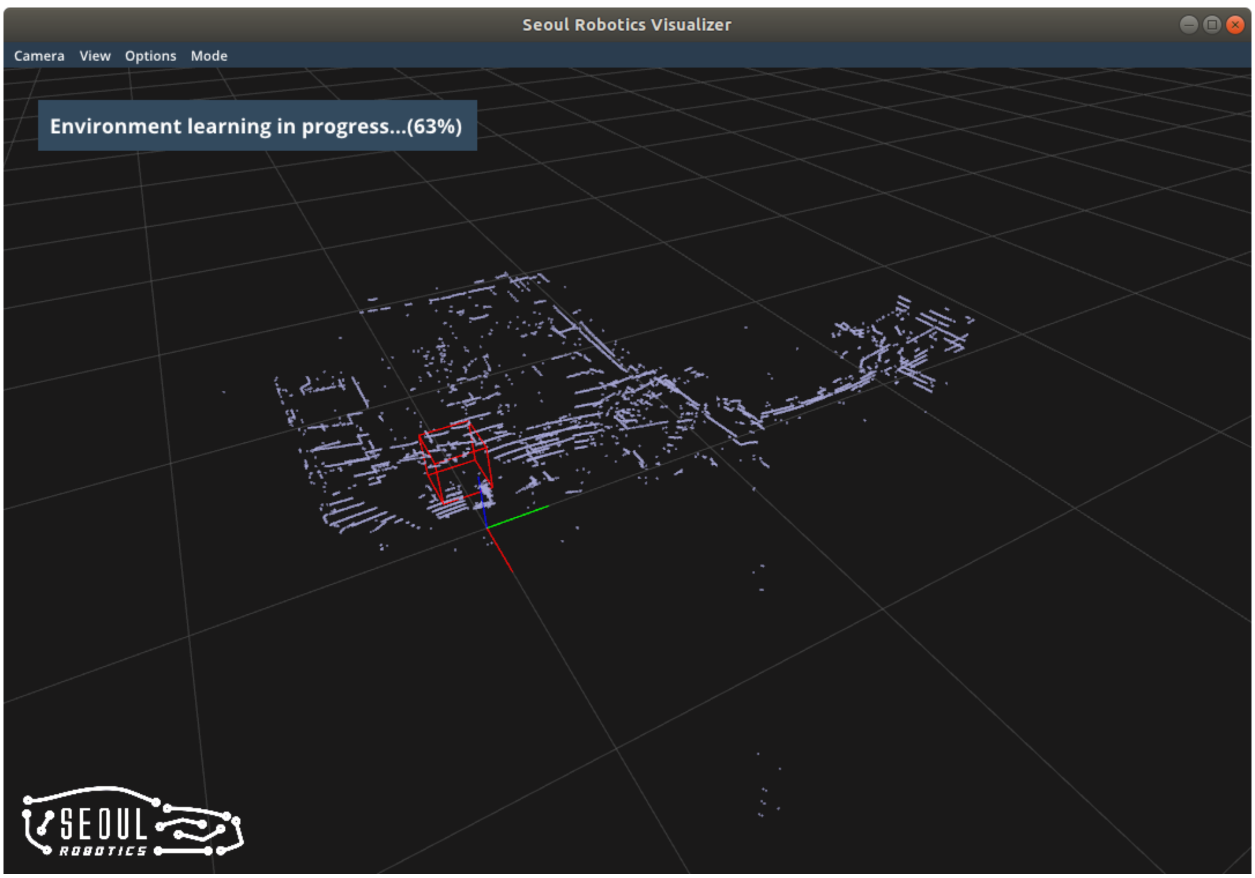
You can visualize a floor plan or a map for the sensor calibration or monitoring in **Mode > Map Setup**. You can load an image in your computer with **Load** button and transform it as you want with sliders. With Ctrl + Click combo, you can edit the values of the sliders directly. When you change the value, that automatically saved. You can also remove an image with **Remove** button.



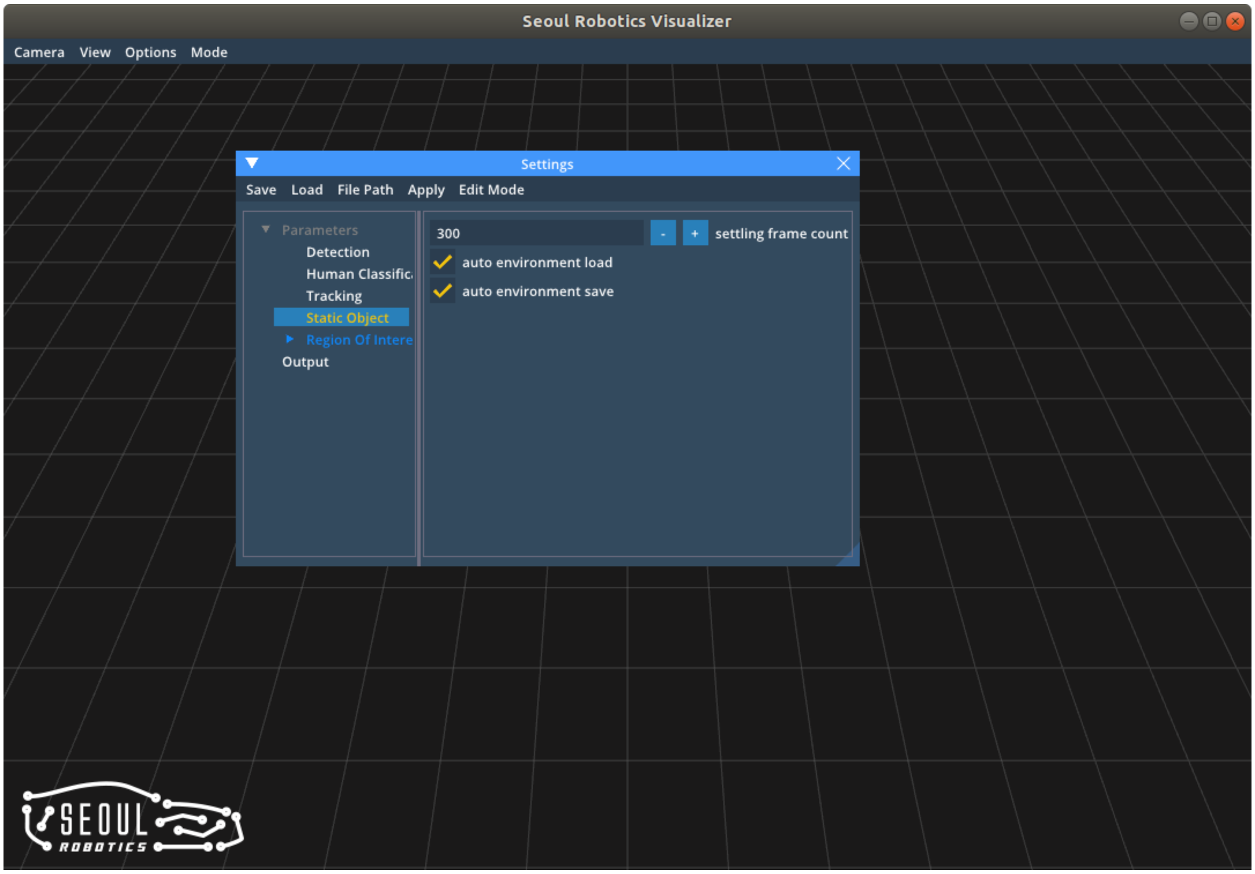
6. Runtime

Step 1: Environment Learning

SENSR runs the initial environment learning before starting object detection. It keeps updating the environment while detecting objects so that it can deal with the environment change such as shifted boxes.

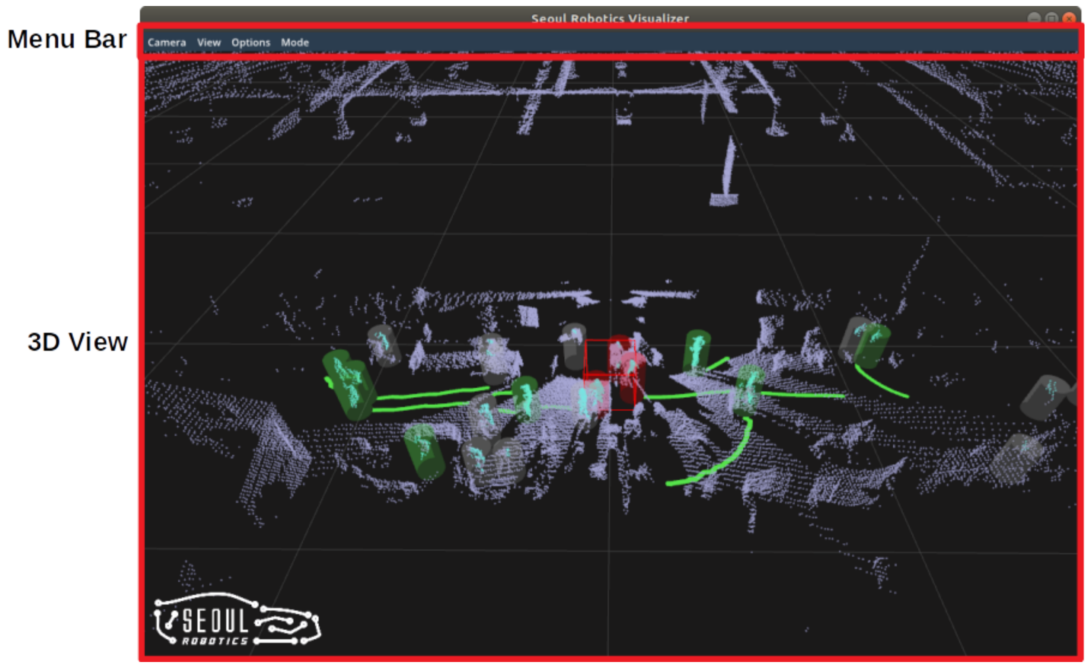


SENSR supports saving and loading the environment automatically, which saves the learning time by reusing the previous environment. They can be configured at Parameters/Static Object in Settings.



Step 2: Runtime

Once the Environment Learning is done, SENSR processes Lidar data to detect, classify and track objects. Input point cloud and result is rendered in 3D View while the output result is published from SENSR to the client. You can find options to get more information or control your settings in the Menu Bar.



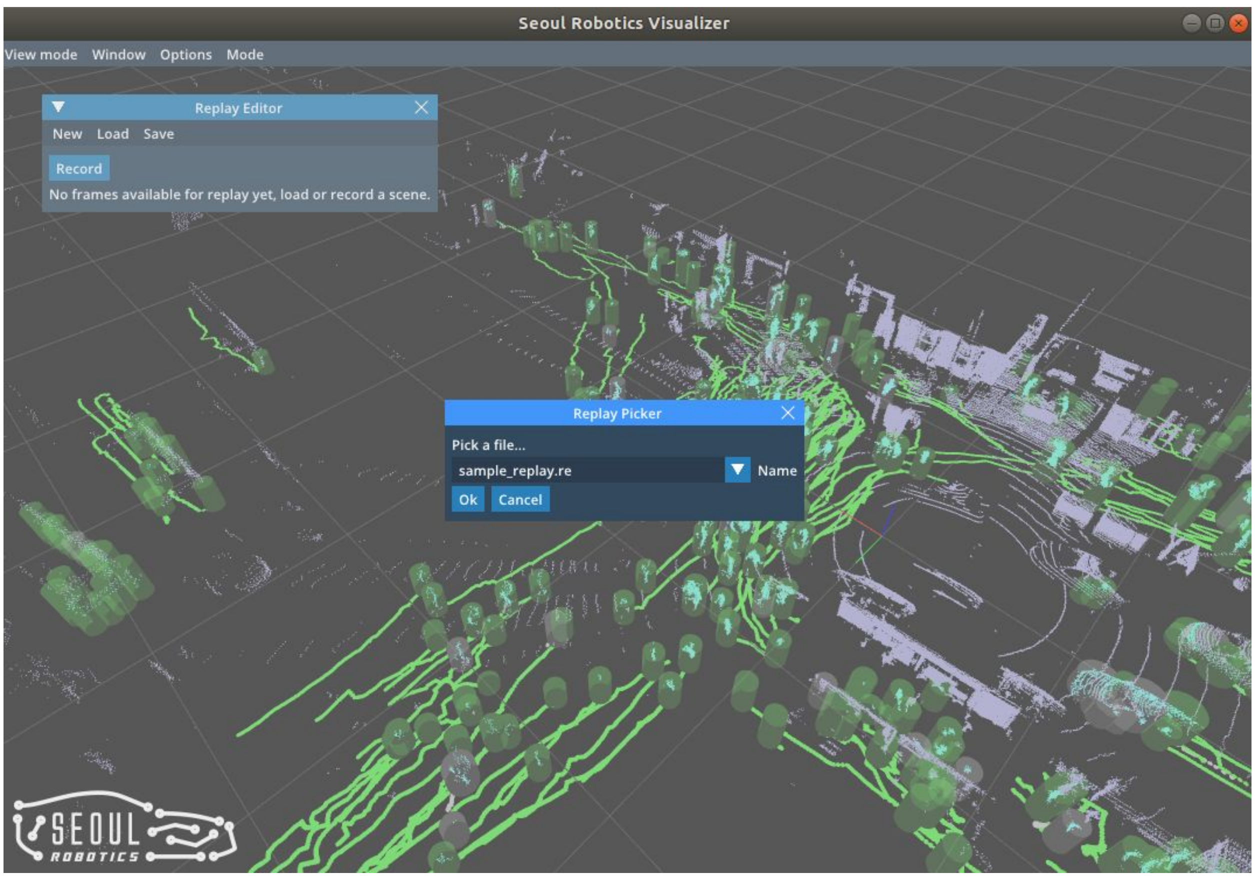
Menu Bar

- The **View mode** tab has several view modes such as Perspective view, Orthographic view (both top and side view) and Isometric view. There is also an option to reset the view to the default.
- The **Windows** tab has additional viewers for supplementary information.
  - The **Object** shows information about tracked objects.
  - The **Sensor Status** view shows information about the sensors currently being used.
  - The **Replay Editor** allows users to record, save and load replays to inspect the output in detail.

- The **Options** tab includes **Enable Rendering** (to turn off rendering for less consumption of computing power), **Settings** (to configure parameters and options), and **Rendering Config Editor** (to configure rendering options for the 3D View).
- The **Mode** tab offers Sensor Setup, Lidar Calibration, Zone Setup, and Map Setup that can be edited at any time.

7. Replay Editor (Optional)

Allows users to record, save and load replays to inspect the output in detail. To open **Window/Replay Editor**. Click record to start and save to save the replay file. The name of the file needs to be provided. The replay file resides in **sensr_i/sensr_i/data/replay**. To open, click **Load** and choose the saved replay file. Note that only replay files in **sensr_i/sensr_i/data/replay** are recognized, using replay file in sample data, copy it to that folder.



8. Settings

To change the settings you can open the **Settings** window (with the **Option > Settings** button in the menu bar). It is recommended to change some parameters (such as detection range) to better fit each specific use case. A full description of the parameter specification can be seen below.
Note that some parameters such as the detection range have to be chosen so that the lower limit is lower than the upper limit. Additionally, all range values are specified w.r.t. the origin point (shown by the xyz-axis in the visualizer). Be advised to change algorithm parameters with caution, as wrong parameters can negatively affect the performance.

Settings Parameter Specification

Location	Name	Description	Unit	Default
Parameters > Detection	X range	X range of detection space (w.r.t. origin).	meters	-50.0, 50.0
	Y range	Y range of detection space (w.r.t. origin).	meters	-50.0, 50.0
	Z range	Z range of detection space (w.r.t. origin).	meters	-1.5, 2.5
	Minimum point count	The minimum number of points of an object to be detected.	-	3
Parameters > Tracking	Num history	The number of history positions to keep.	-	100
	Drifting period	Period of short term prediction when an object disappears.	seconds	1.0
Parameters > Static Object	Settling frame count	The number of Lidar input frames to estimate a static object as environment.	-	30
	Auto environment load	Load environment learnt in the previous execution	-	false
	Auto environment save	Save learnt environment at the end of the application	-	false

Parameters > Region Of Interest > Zone	Id	ID of the zone.	-	-
	Minimum Z	Minimum height of the zone (w.r.t. origin).	meters	-
	Maximum Z	Maximum height of the zone w.r.t. origin).	meters	-
	Zone Type	Type of the zone, which can be either 0 (Exclusion Zone) or 1 (Event Zone)	-	-
	Vertices	List of the XY vertices of the zone.	meters	-
Output	Publish Point Cloud	Publish point cloud X through output if enabled.	-	true

Rendering Config Editor

Location	Name	Description	Default
Visibility	Object Points	Enable or disable showing points inside objects. (e.g. points inside Pedestrian)	True
	Ground Points	Enable or disable showing ground points. (e.g. floor..)	True
	Environment Points	Enable or disable showing environment points. (e.g. wall or desk…)	True
	Grid	Enable or disable showing squared Grid.	True
	Circular Grid	Enable or disable showing circular Grid.	False
	Axes	Enable or disable showing axes in the center of view.	True
	Objects	Enable or disable showing tracked objects. (e.g. pedestrian)	True
	Misc Objects	Enable or disable showing miscellaneous objects.	True
	2D Map Image	Enable or disable showing a map image.	True
View Range	Min point height	Minimum height to show points (Unit: meters) The point lower than this value will not be shown even if you enabled Env. Points ON.	-10.0
	Max point height	Maximum height to show points (Unit: meters) The point higher than this value will not be shown even if you enabled Env. Points ON.	10.0

9. REST API

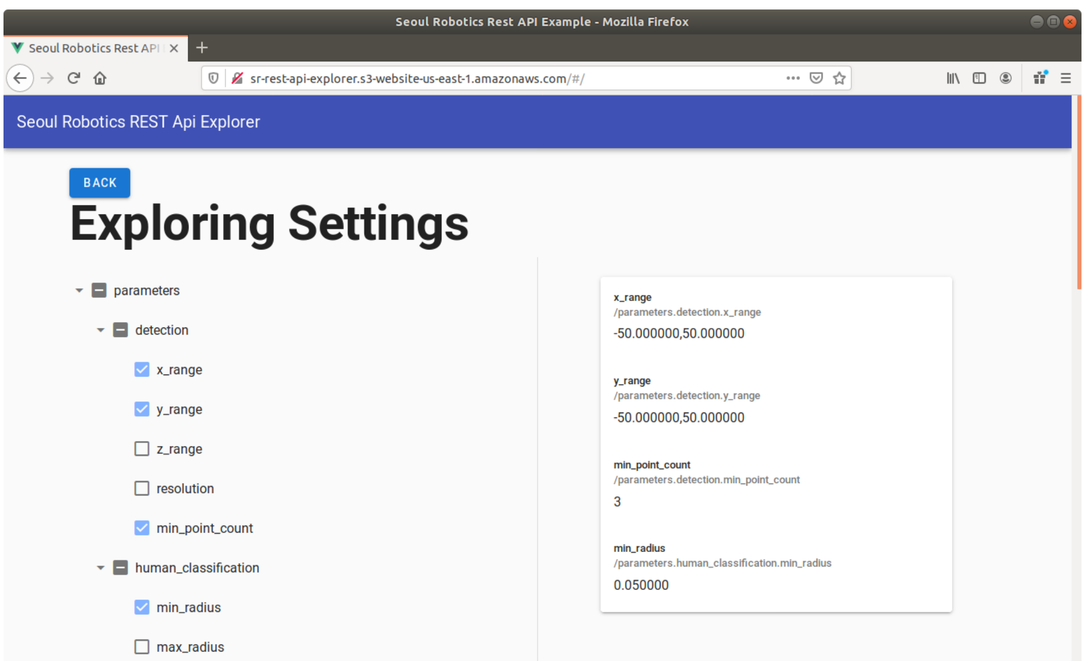
SENSR can be remote controlled by REST API. You can change the parameters, add and remove zones, and restart the pipeline with REST API.

Specification

Full documentation of REST API specification including usage example is here: http://sr-rest-api-docs.s3-website.ap-northeast-2.amazonaws.com.

REST API Explorer

For debugging purposes, you can use a web interface to explore your parameters. Run your SENSR first and check your parameters here: http://sr-rest-api-explorer.s3-website-us-east-1.amazonaws.com.



10. Output

SENSR encodes data processing results with Protobuf and sends it through TCP using ZeroMQ. With SENSR SDK, you can easily receive SENSR output in your C++ client. SENSR SDK code is here: https://github.com/seoulrobotics/sensr_sdk/tree/v1.2.0.

Communication Specification

Communication Protocol	TCP (ZeroMQ)
Data Encoding	Protobuf 3
Output Port	5050

Output Data Specification

Output Protobuf code is available here:

https://github.com/seoulrobotics/sensr_sdk/blob/v1.2.0/proto/output.proto

1. Output Message Specification

Field	Description	Contents
time_stamp	Timestamp of the LiDAR input message.	- Second - Nanosecond (decimal value)
ground_points	List of ground points in calibrated coordinate system.	- XYZ points of the ground
objects	Information of detected objects	- Object ID - Bounding box of the object - Label (Classification) - Classification probability - Boolean flag of tracking reliability - Velocity of the object in world coordinate - XYZ points of history (trace) - XYZ points of predicted object positions
region_of_interest	Information of region of interest zones	- Zone ID - Polygon box of the region of interest - List of object IDs found in the zone - Boolean flag if the zone is an exclusion zone or not

2. Message Format (Protobuf) Specification

Message	Field	Data Type	Description	Unit
Timestamp	seconds	int64	Timestamp in seconds.	seconds
	nanos	int32	Timestamp decimal value in nanoseconds.	nano seconds
Vector2	x	float	X value of the 2D vector.	meters
	y	float	Y value of the 2D vector.	meters
Vector3	x	float	X value of the 3D vector.	meters
	y	float	Y value of the 3D vector.	meters
	z	float	Z value of the 3D vector.	meters
BoundingBox	position	Vector3	XYZ position of the bounding box. Position is defined as (Center X, Center Y, Bottom Z).	meters
	size	Vector3	XYZ size of the bounding box.	meters
	yaw	float	Bounding box rotation angle along the Z axis.	radians
PolygonBox	points	Vector2 (list)	List of XY vertices of the polygon box.	meters
	min_z	float	Minimum Z of the box.	meters
	max_z	float	Maximum Z of the box.	meters
TrackingInfo	probability	float	0 to 1 probability of the object classification.	-
	tracking_reliable	bool	True if the object’s tracking is reliable.	-
	velocity	Vector3	XYZ velocity of the object.	m/s
	history	Vector3 (list)	List of bounding box’s XYZ position values along the tracking history.	meters
	prediction	Vector3 (list)	List of bounding box’s XYZ position values in the future.	meters
Object	id	int32	ID of the object.	-
	bbox	BoundingBox	Bounding box of the object	-

	label	int32	Classification of the object. 0: Car, 1: Pedestrian, 2: Cyclist, 3: Misc	-
	track	TrackingInfo	Tracking information of the object	-
	points	bytes	List of sequential 3 float value set (XYZ).	meters
Zone	id	int32	ID of the zone	-
	polygon_box	PolygonBox	Polygon box-shaped volumetric range of the zone.	-
	object_ids	int32 (list)	List of object IDs found in the zone	-
	zone_type	int32	Type of the zone	-

Revision and Release History

Revision	Release date	Description
A0.0.0	2019.08.09	First release
A0.0.1	2019.08.13	Description update for calibration
A0.0.2	2019.08.19	Dependency and Contents outline update
A0.0.3	2019.08.20	Remove invalid points (v1.0.3)
A0.1.0	2019.09.06	Update system setup, settings and output (v1.1.0 beta)
A0.1.1	2019.10.25	1.1 Full support (v1.1.0)
A0.1.2	2020.02.12	1.2 Full Support (v1.2.0)