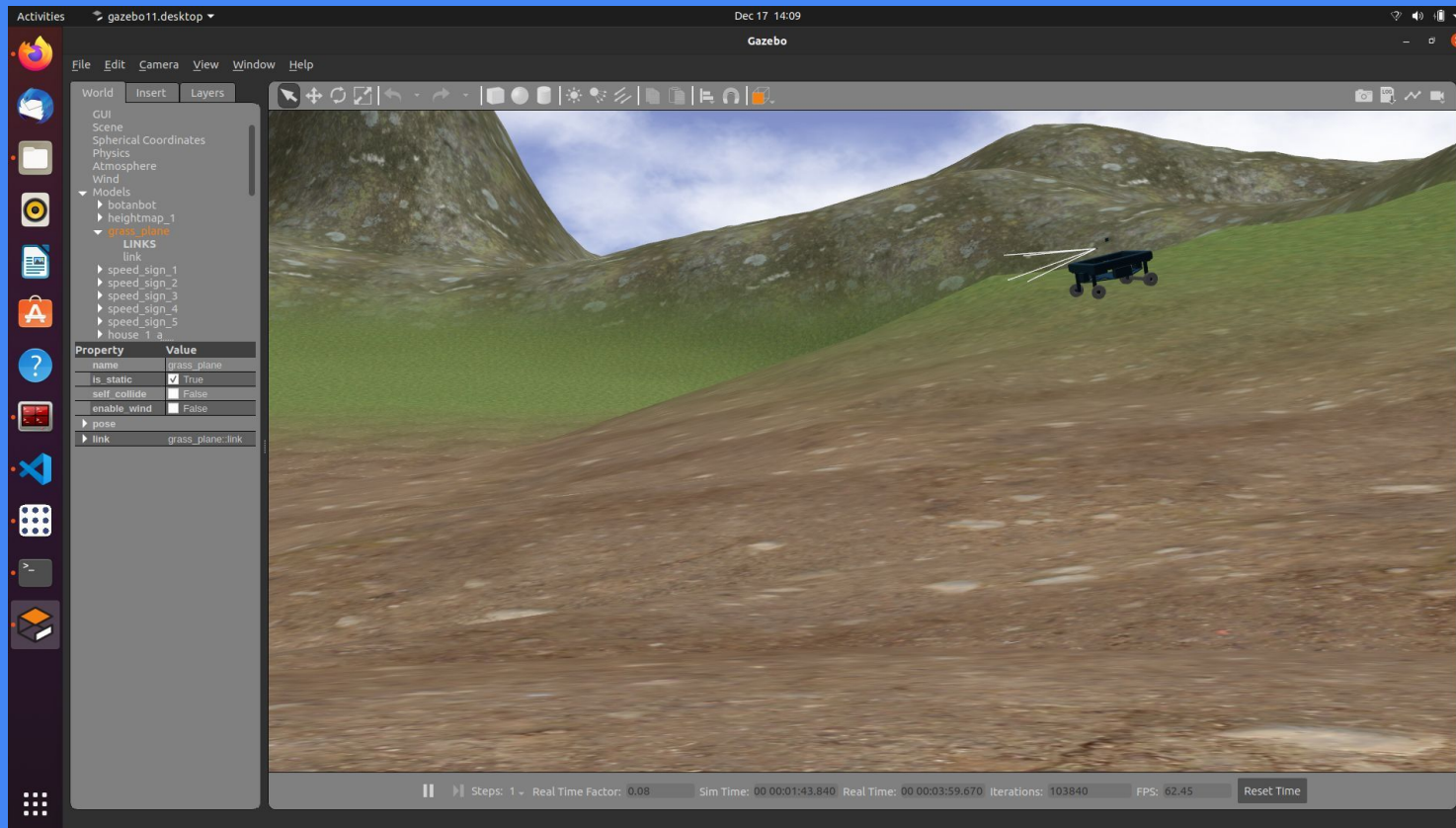


Navigation on Uneven Terrain



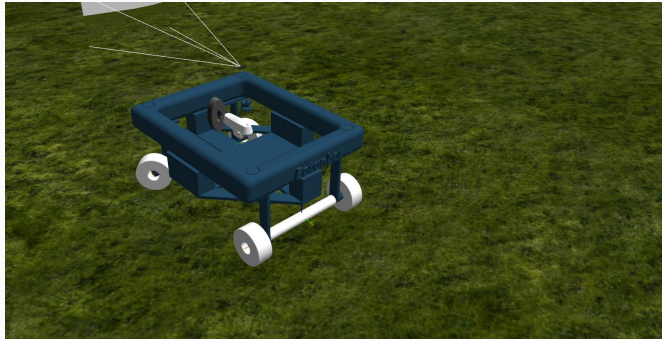
Motivation

Given a robot decorated with all required sensors, autonomously traverse a uneven field, with minimal overlapping of routes.

- Agricultural fields aren't flat, ramps, bumbs , elevations are in nature of agri-fields
- Think of a way that is not limited to 2D Occupancy grid for navigation.

Botanbot

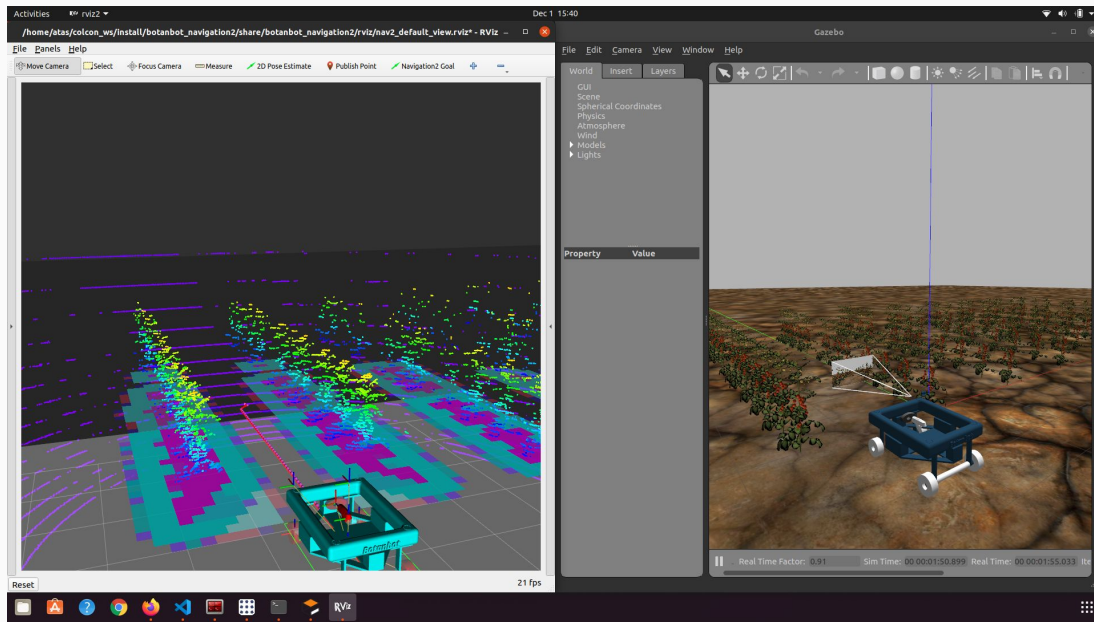
Botanbot is a simple 4 wheeled , ackermann driven mobile robot. It is simulated under Gazebo with all required essential sensors in order to do **outdoor navigation**. The following table shows currently supported sensors.



Sensor Type	Topic	MSG TYPE
LIDAR	/velodyne_points	sensor_msgs::msg::
RealSense D435 COLOR CAMERA	/camera/color/image_raw	sensor_msgs::msg::
RealSense D435 DEPTH CAMERA	/camera/ aligned_depth_to_color/ image_raw	sensor_msgs::msg::
RealSense D435 IR1 CAMERA	/camera/infra1/image_raw	sensor_msgs::msg::
RealSense D435 IR2 CAMERA	/camera/infra2/image_raw	sensor_msgs::msg::
GPS	/gps/fix	sensor_msgs::msg::
IMU	/imu/data	sensor_msgs::msg::

Botanbot

Botanbot in a farming simulation environment



Botanbot Software stack

[https://github.com/jediofgever/
OUTDOOR_NAV2](https://github.com/jediofgever/OUTDOOR_NAV2)

[https://outdoor-nav2.readthedocs
io/en/main/index.html](https://outdoor-nav2.readthedocs.io/en/main/index.html)

foxy ▾


2 branches

0 tags


















Go to file

Add file ▾

Code ▾

 **jediofgever** introduce botanbot_map_server

413843c 2 hours ago 186 commits

	assets	add presentation	9 days ago
	botanbot_archived_packages	Add NMBU too all licence notifications	2 days ago
	botanbot_description	botanbot_gazebo sensor plugin fixes	23 hours ago
	botanbot_gazebo	cpplinter happy	23 hours ago
	botanbot_gps_waypoint_follower	Add NMBU too all licence notifications	2 days ago
	botanbot_gui	cpplinter happy	23 hours ago
	botanbot_localization	Add NMBU too all licence notifications	2 days ago
	botanbot_map_server	introduce botanbot_map_server	2 hours ago
	botanbot_msgs	attempt to make planner server work	3 days ago
	botanbot_navigation2	Add NMBU too all licence notifications	2 days ago
	botanbot_opencvslam	introduce botanbot_map_server	2 hours ago
	botanbot_planning	cpplinter happy	23 hours ago
	botanbot_utilities	lint all project	23 hours ago
	.gitignore	Add utility package for botanbot	2 months ago
	LICENSE	Initial commit	2 months ago
	README.md	add presentation	9 days ago
	uncrustify.cfg	correct licenses added, introduce gps waypoint flw	2 months ago

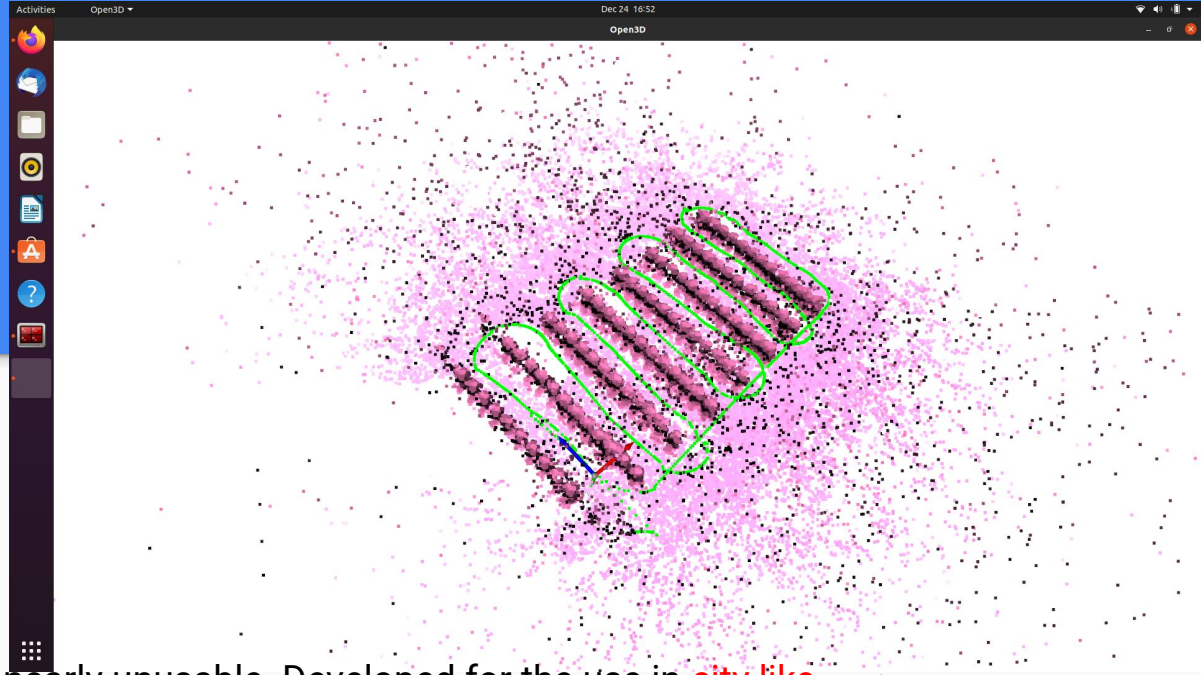
3D global representation of environment

- Realistic representation with reliable collision avoidance and path planning
- Surface models that can support a traversability analysis before robot operates
- Possibility of increase in localization accuracy by 3D scan matchers or VSLAM landmark recognition

Considerations for 3D environment representation?

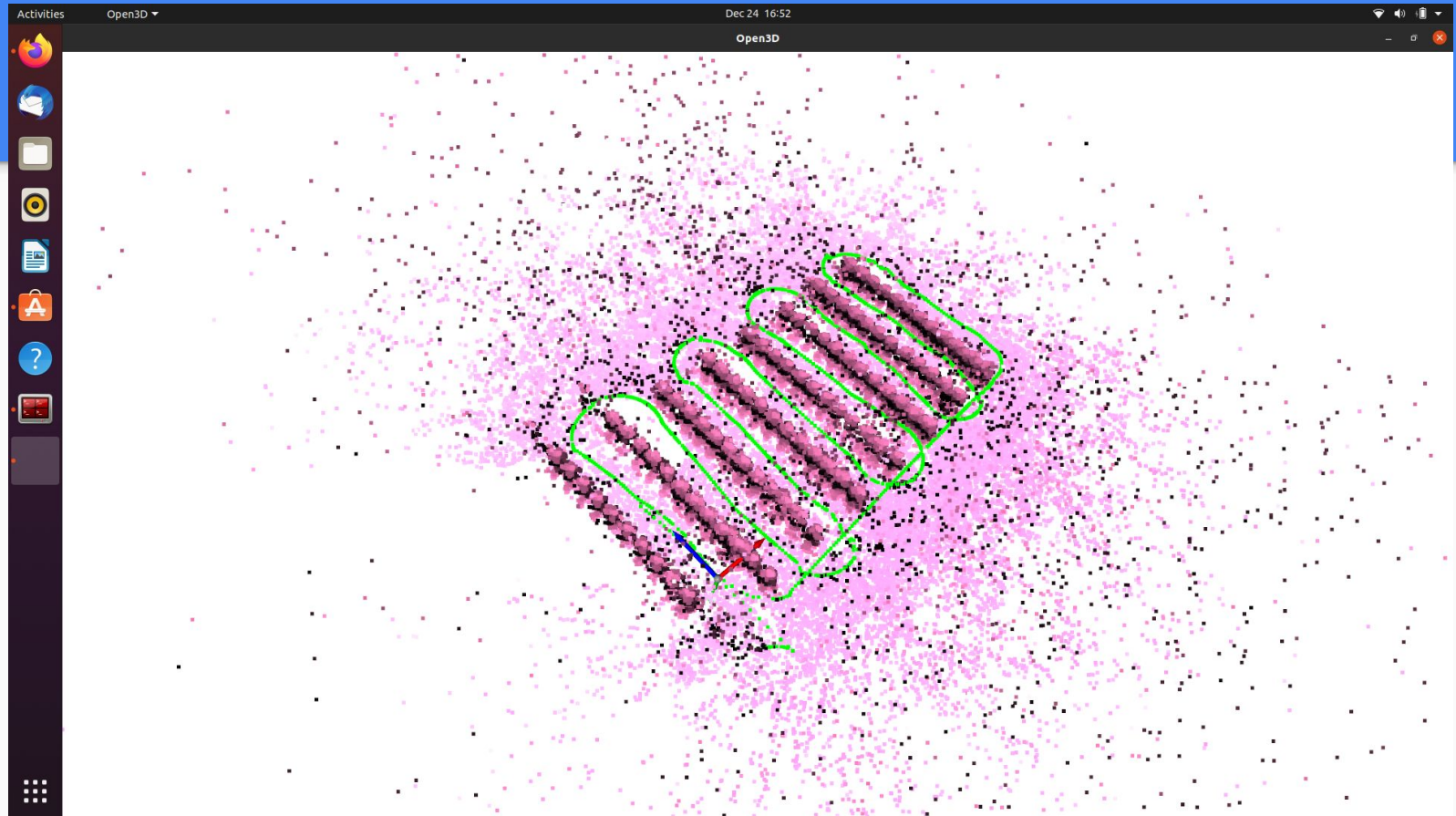
- [Grid_map](#) (2.5D actually, but various layers to embed all 3D data, no integrated planner that I know, but they can be converted to costmap_2d easily)
- [Mesh_map](#) (triangular meshes, have compatible planners and controllers)
- [elevation_mapping](#) (robot-centric or local tasks only, bases on grid_map)
- [traversability_mapping](#) (not a serious consideration but nice to keep it here, based on Lego-LOAM)

SLAM's role

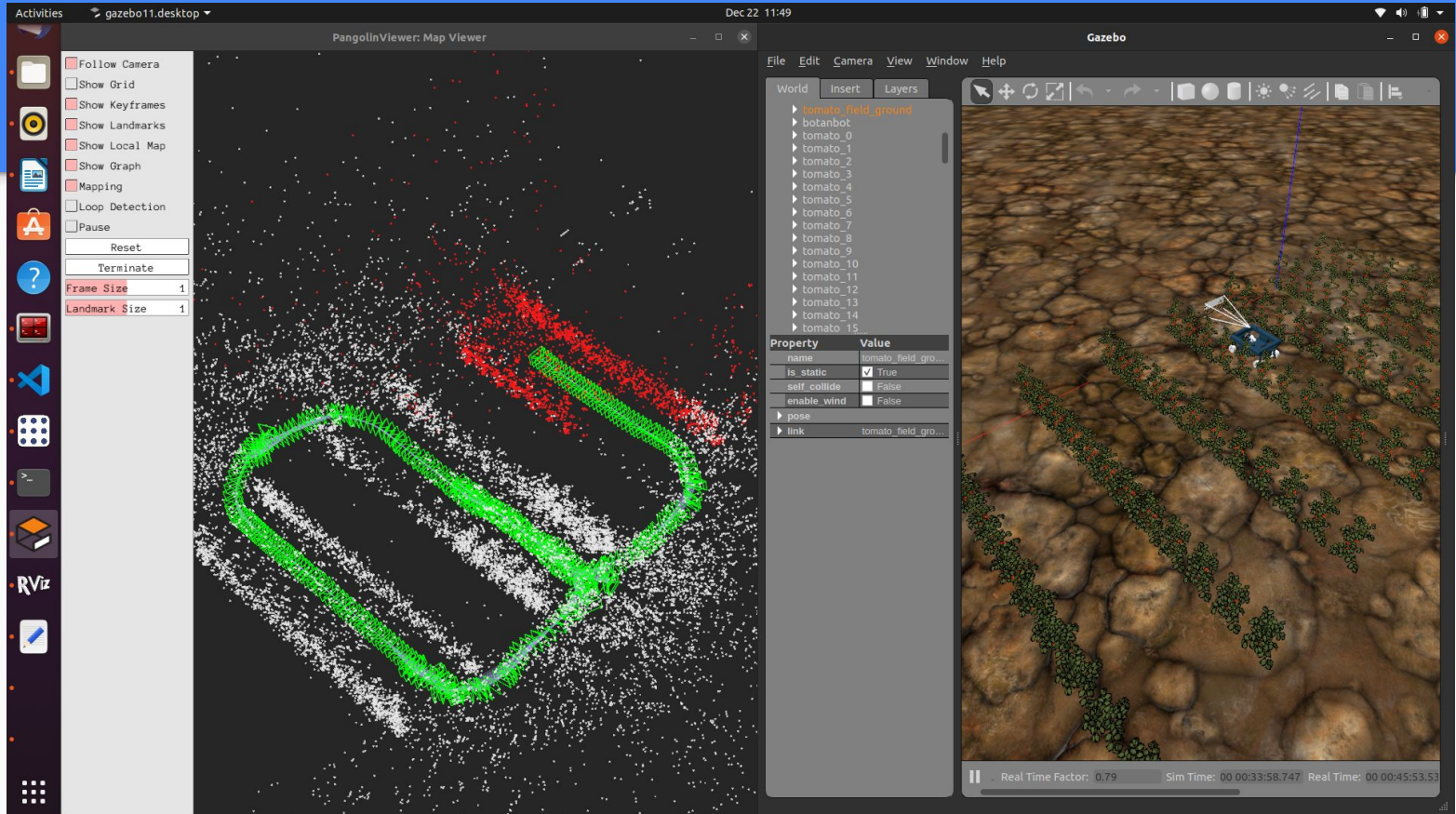


- LIDAR based 3D SLAM in the field is nearly unusable. Developed for the use in **city like** environments.
- Lack of solid shapes, **repetitive occuring** patterns confuses the algorithms.
- Visual SLAM performs better in the field. With an RGBD camera a **correctly scaled map** of agricultural field can be obtained.
- [Botanbot openvslam](#), is a client package to [openvslam](#), a visual slam based o ORB-SLAM2 Both **mapping** and **localization** nodes are available

SLAM's role

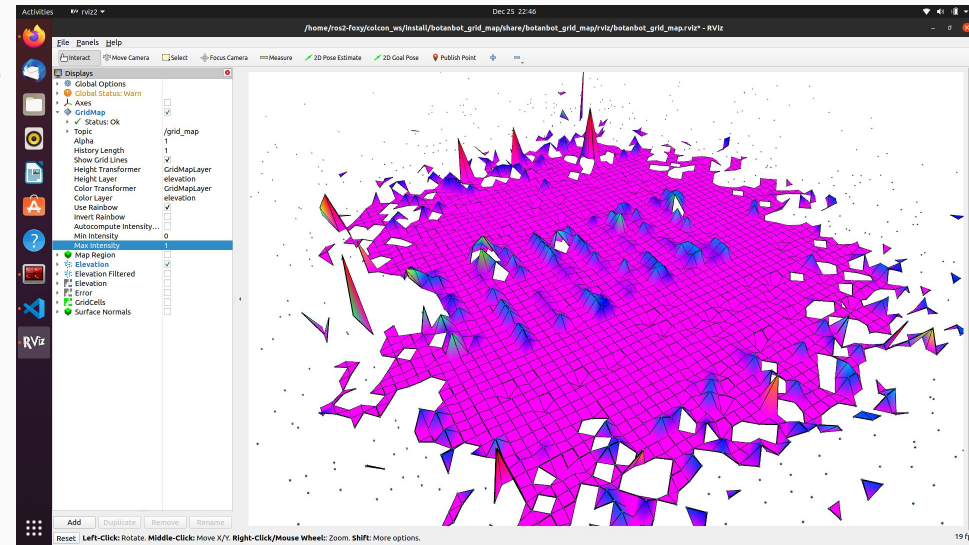


SLAM's role



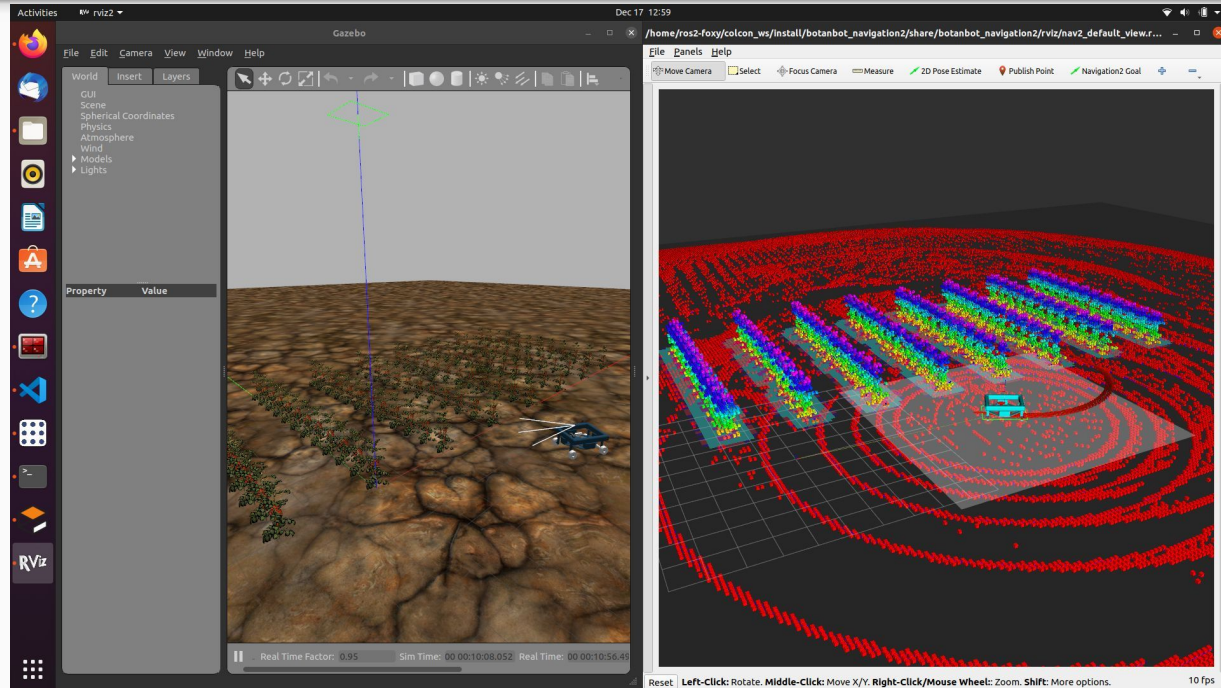
Botanbot Grid Map

- [Botanbot_grid_map](#) is a client package for grid_map, with provided ROS node and config files, a map(.pcd file) created by visual SLAM is retrieved as grid_map and published.
- No direct integration with planners



Botanbot Map Server

- [Botanbot_map_server](#) is under development. It will serve to `botanbot_planning` and rely on `botanbot_openvslam`
- Botanbot Map server uses OCTOMAP which works exceptionally well for `collision checking` while planning with OMPL planners



How to do navigation based on this 3D map?

- [Botanbot_planning](#) is a package that provides plugin based structure, plugins access to planners available in OMPL
- With [Flexible Collision Library](#), collision check between robot's skeleton and map is performed. The map is represented as **octree** and robot body as 3D bounding box.
- An action interface to do planning is provided as `ComputePathToPose`;

ComputePathToPose.action interface

```
#goal definition

geometry_msgs/PoseStamped pose

string planner_id

---

#result definition

geometry_msgs/PoseStamped[] path

builtin_interfaces/Duration planning_time

#feedback

builtin_interfaces/Duration elapsed_time
```

Available planner plugins are
three now

```
planner_plugin: "SE2PlannerControlSpace" # other options: "SE2Planner", "SE3Planner", "SE2PlannerControlSpace"
expected_planner_frequency: 5.0
SE2Planner:
  plugin: "botanbot_planning::SE2Planner"
  planner_name: "RRTStar" # other options: RRTStar, RRTConnect, KPIECE1
  planner_timeout: 1.0
  interpolation_parameter: 50
  octomap_filename: "/home/ros2-foxy/f.bt"
  octomap_voxel_size: 0.1
  state_space_boundaries:
    minx: -50.0
    maxx: 50.0
    miny: -50.0
    maxy: 50.0
    minz: 0.0
    maxz: 3.0
  robot_body_dims:
    x: 1.0
    y: 0.5
    z: 0.4
```

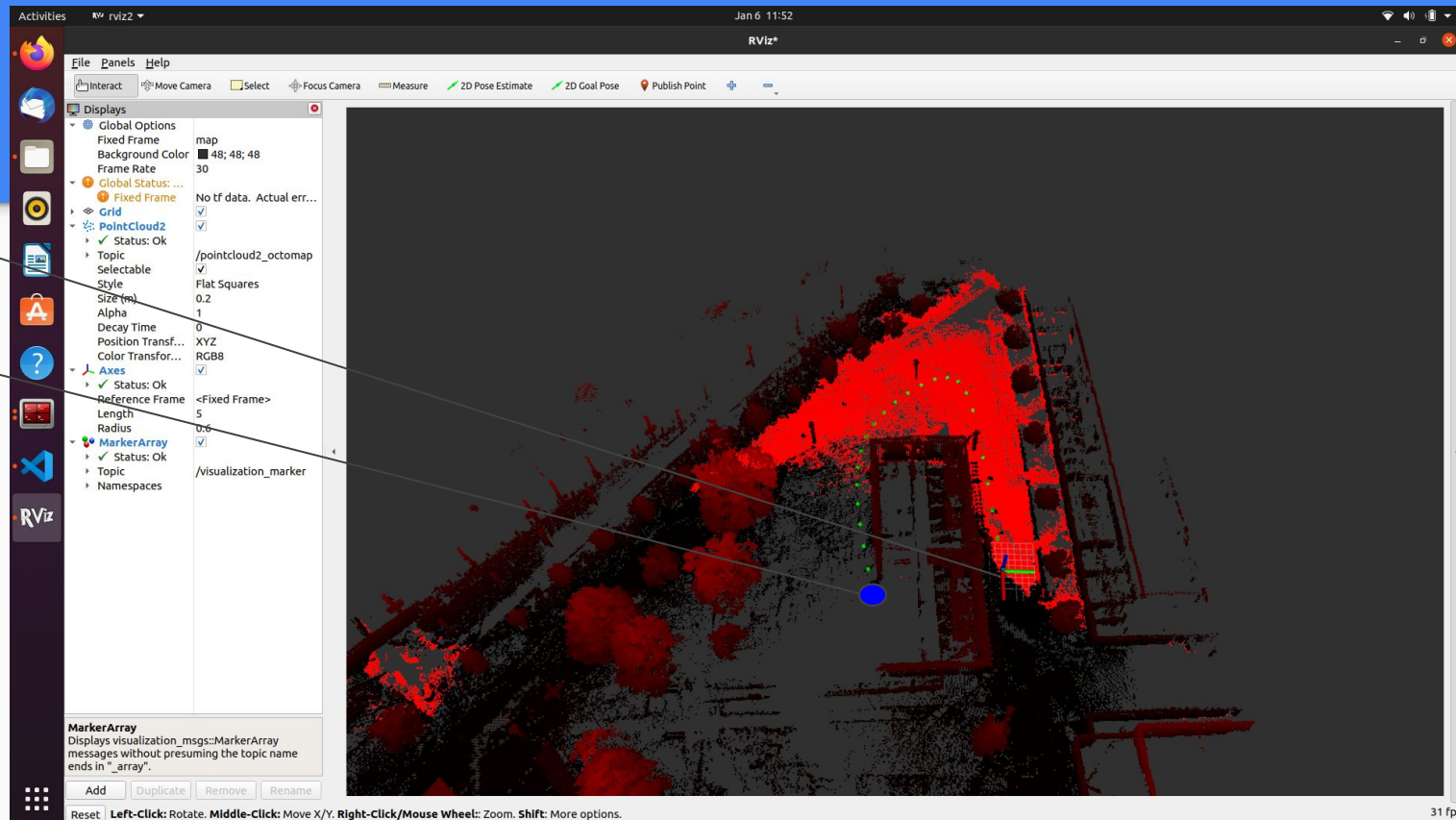
```
SE2PlannerControlSpace:
  plugin: "botanbot_planning::SE2PlannerControlSpace"
  planner_name: "RRT" # other options: SST, EST, KPIECE1
  planner_timeout: 1.0
  interpolation_parameter: 50
  octomap_filename: "/home/ros2-foxy/f.bt"
  octomap_voxel_size: 0.1
  state_space_boundaries:
    minx: -50.0
    maxx: 50.0
    miny: -50.0
    maxy: 50.0
    minz: 0.0
    maxz: 3.0
  robot_body_dims:
    x: 1.0
    y: 0.5
    z: 0.4
```

```
SE3Planner:
  plugin: "botanbot_planning::SE3Planner"
  planner_name: "PRMStar" # other options: PRMStar, RRTConnect, KPIECE1
  planner_timeout: 1.0
  interpolation_parameter: 50
  octomap_filename: "/home/ros2-foxy/f.bt"
  octomap_voxel_size: 0.1
  state_space_boundaries:
    minx: -50.0
    maxx: 50.0
    miny: -50.0
    maxy: 50.0
    minz: 0.0
    maxz: 3.0
  robot_body_dims:
    x: 1.0
    y: 0.5
    z: 0.4
```


Some results of planning in 3D

Start(0.0, 0.0, 1.0)

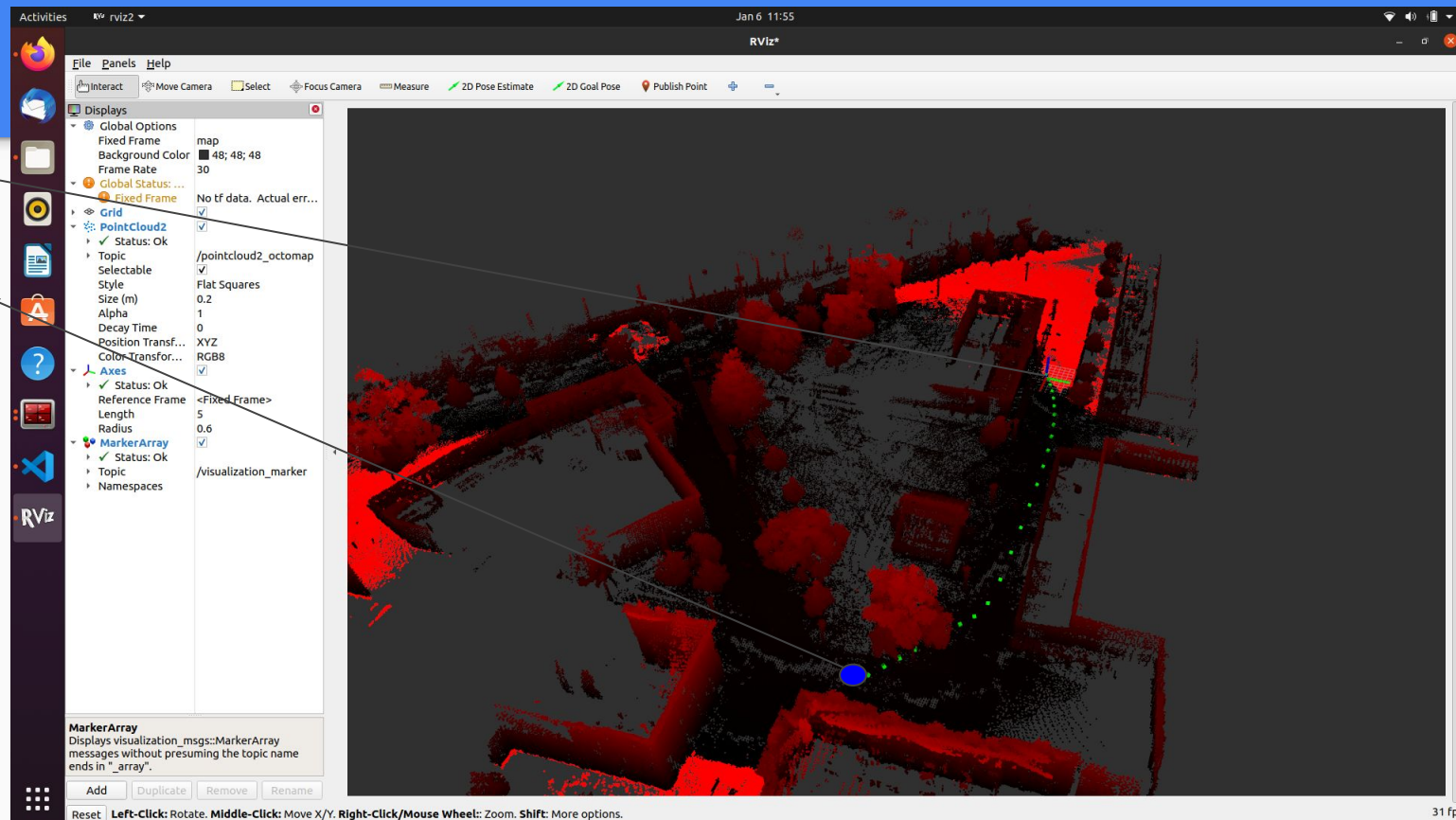
Goal(5.0, -25.0, 1.0)



Some results of planning in 3D

Start(0.0, 0.0, 1.0)

Goal(80.0, -5.0, 1.0)



Localization (WIP)

Rework botanbot_localization so that we have following;

- Absolute localization (utm -> base_link) GNSS + **relative_localization** = absolute_localization
- Relative localization (map -> base_link) **local_localization** + VSLAM = **relative_localization**
- Local localization (odom->base_link) vslam+imu+encoder = **local_localization**

Botanbot_localization currently does not include map built by SLAM into localization, it only uses GNSS, IMU, Wheel Odometry.

,current effort will be integrating a built map into botanbot_localization.

Localization (WIP)

- When doing the SLAM in mapping mode, write map info with following properties.
- With this info, **utm -> map** can be received.

```
1 sensor_type: rgbd
2 map_db_path: /home/ros2-foxy/f.msg
3 creation_date: "Fri Jan 15 11:34:42 2021\n"
4 map_coordinates:
5   latitude: 49.89998651126992
6   longitude: 8.90004175541922
7   altitude: 0.6342219080730283
8   quaternion:
9     x: -0.0002674302377596963
10    y: -7.804134265664371e-05
11    z: 0.7068408908205831
12    w: 0.7073725167509558
```

Origin of map



Thanks!