rtabmap (/rtabmap): *rtabmap (/rtabmap)* | *rtabmap_ros*

indigo	kinetic	lunar	melodic	*Show EOL distros:* □

Documentation Status

Package Links

- **Code API (http://docs.ros.org/melodic/api/rtabmap_ros/html)**
- **Msg/Srv API (http://docs.ros.org/melodic/api/rtabmap_ros/html/index-msg.html)**
- Tutorials (/rtabmap_ros/Tutorials)
- FAQ (http://answers.ros.org/questions/scope:all/sort:activity-desc/tags:rtabmap_ros/page:1/)
- Change List (/rtabmap_ros/ChangeList)
- Reviews (/rtabmap_ros/Reviews)

Dependencies (35)
Jenkins jobs (9)

Package Summary

✓ **Released** ✓ **Continuous Integration** ✓ **Documented**

RTAB-Map's ros-pkg. RTAB-Map is a RGB-D SLAM approach with real-time constraints.

- Maintainer status: maintained
- Maintainer: Mathieu Labbe <matlabbe AT gmail DOT com>
- Author: Mathieu Labbe
- License: BSD
- Bug / feature tracker: https://github.com/introlab/rtabmap_ros/issues (https://github.com/introlab/rtabmap_ros/issues)
- Source: git https://github.com/introlab/rtabmap_ros.git (https://github.com/introlab/rtabmap_ros) (branch: melodic-devel)

Contents

1. Overview

This package is a ROS wrapper of RTAB-Map (http://introlab.github.io/rtabmap) (Real-Time Appearance-Based Mapping), a RGB-D SLAM approach based on a global loop closure detector with real-time constraints. This package can be used to generate a 3D point clouds of the environment and/or to create a 2D occupancy grid map for navigation. The tutorials (http://wiki.ros.org/rtabmap_ros#Tutorials) and demos (http://wiki.ros.org/rtabmap_ros#Demos) show some examples of mapping with RTAB-Map.

2. Tutorials

1. RGB-D Handheld Mapping (/rtabmap_ros/Tutorials/HandHeldMapping)
   This tutorial shows how to use rtabmap_ros (/rtabmap_ros) out-of-the-box with a Kinect-like sensor in mapping mode or localization mode.
2. Stereo Handheld Mapping (/rtabmap_ros/Tutorials/StereoHandHeldMapping)
   This tutorial shows how to use rtabmap_ros (/rtabmap_ros) out-of-the-box with a stereo camera in mapping mode or localization mode.
3. Remote Mapping (/rtabmap_ros/Tutorials/RemoteMapping)
   This tutorial shows how to do mapping on a remote computer.
4. Stereo Outdoor Mapping (/rtabmap_ros/Tutorials/StereoOutdoorMapping)
   This tutorial shows how to do stereo mapping with RTAB-Map.
5. Stereo Outdoor Navigation (/rtabmap_ros/Tutorials/StereoOutdoorNavigation)
   This tutorial shows how to integrate autonomous navigation with RTAB-Map in context of outdoor stereo mapping.
6. Setup RTAB-Map on Your Robot! (/rtabmap_ros/Tutorials/SetupOnYourRobot)
   This tutorial shows multiple RTAB-Map configurations that can be used on your robot.
7. Mapping and Navigation with Turtlebot (/rtabmap_ros/Tutorials/MappingAndNavigationOnTurtlebot)
   This tutorial shows how to use RTAB-Map with Turtlebot for mapping and navigation.
8. Tango ROS Streamer (/rtabmap_ros/Tutorials/Tango%20ROS%20Streamer)
   Tutorial to get Tango ROS Streamer working with rtabmap_ros
9. Advanced Parameter Tuning (/rtabmap_ros/Tutorials/Advanced%20Parameter%20Tuning)
   This tutorial tells you which parameter to change to improve performances
10. Wifi Signal Strength Mapping (User Data Usage) (/rtabmap_ros/Tutorials/WifiSignalStrengthMappingUserDataUsage)
    This tutorial shows how to add user data during mapping that will be saved directly in RTAB-Map's database for convenience.

3. Demos

3.1 Robot mapping

For this demo, you will need the ROS bag demo_mapping.bag (https://docs.google.com/uc?id=0B46akLGdg-uadXhLeURiMTBQU28&export=download) (295 MB, **fixed camera TF 2016/06/28**, **fixed not normalized quaternions 2017/02/24**).

Launch: demo_robot_mapping.launch (https://github.com/introlab/rtabmap_ros/blob/master/launch/demo/demo_robot_mapping.launch)

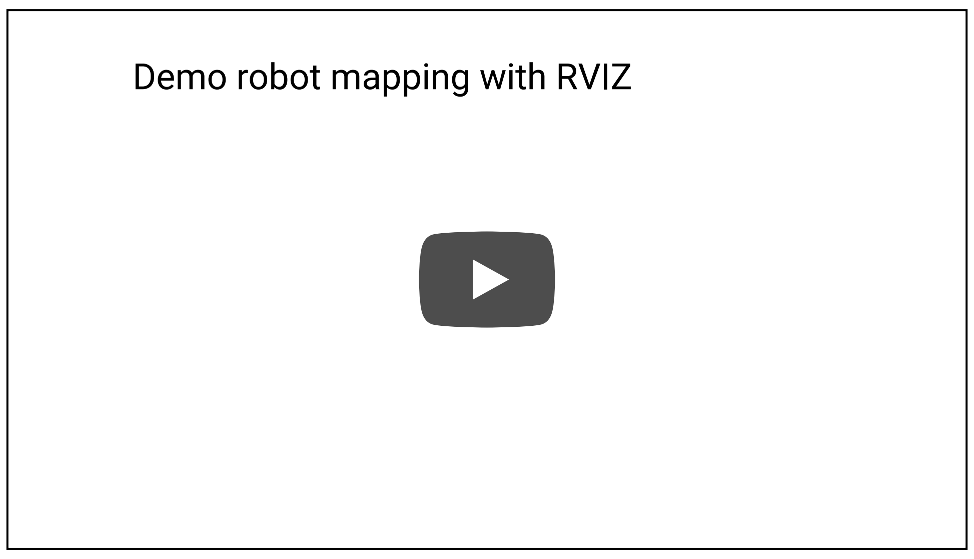
```
$ roslaunch rtabmap_ros demo_robot_mapping.launch
$ rosbag play --clock demo_mapping.bag
```

After mapping, you could try the localization mode:

```
$ roslaunch rtabmap_ros demo_robot_mapping.launch localization:=true
$ rosbag play --clock demo_mapping.bag
```

- Note that the GUI node doesn't download automatically the map when started. You will need to click "Edit->Download Map" on the GUI to download the map from the core node.

3.2 Robot mapping with RVIZ



For this demo, you will need the ROS bag demo_mapping.bag (https://docs.google.com/uc?id=0B46akLGdg-uadXhLeURiMTBQU28&export=download) (295 MB, **fixed camera TF 2016/06/28**, **fixed not normalized quaternions 2017/02/24**).

Launch: demo_robot_mapping.launch (https://github.com/introlab/rtabmap_ros/blob/master/launch/demo/demo_robot_mapping.launch)

```
$ roslaunch rtabmap_ros demo_robot_mapping.launch rviz:=true rtabmapviz:=false
$ rosbag play --clock demo_mapping.bag
```

3.3 Multi-session mapping

Detailed results are shown on the Multi-session (https://github.com/introlab/rtabmap/wiki/Multi-session) page on RTAB-Map's wiki.

For this demo, you will need the ROS bags of five mapping sessions:

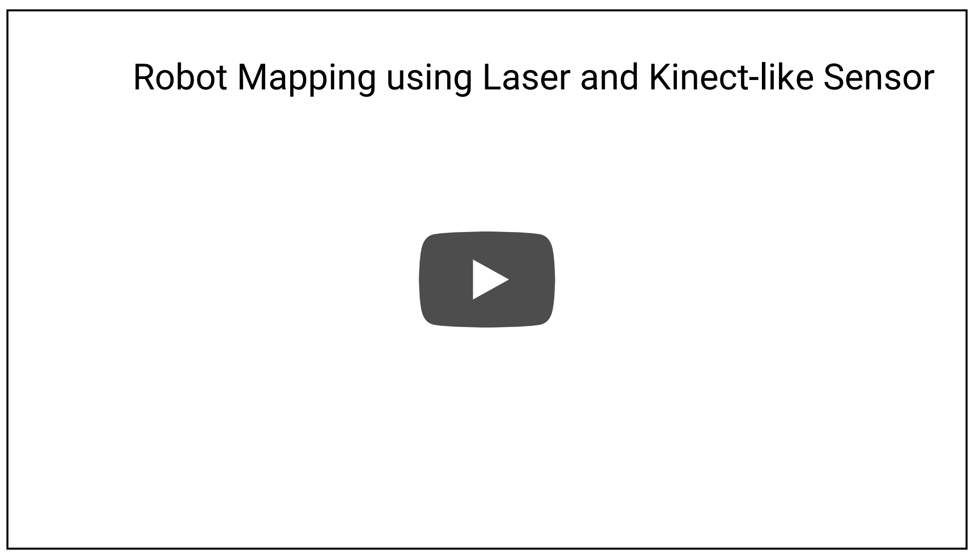
- **Updated 2015/09/21**: fixed TF rotation of the camera
- **Updated 2017/09/08**: fixed TF quaternion not normalized error
- map1.bag.zip (https://drive.google.com/uc?export=download&confirm=wwQS&id=0B46akLGdg-uaa1dDSlUwWUsyTzQ) (542 MB)
- map2.bag.zip (https://drive.google.com/uc?export=download&confirm=nSzj&id=0B46akLGdg-uaUDNKT0RFYl90ejQ) (490 MB)
- map3.bag.zip (https://drive.google.com/uc?export=download&confirm=m6u1&id=0B46akLGdg-uaUzhpRTFfSW1nRk0) (318 MB)
- map4.bag.zip (https://drive.google.com/uc?export=download&confirm=RNlz&id=0B46akLGdg-uaQ0doVVFKYTk1SGM) (1.1 GB)
- map5.bag.zip (https://drive.google.com/uc?export=download&confirm=QDwm&id=0B46akLGdg-uaY2FDQTZ2dzNrMTA) (843 MB)

For the first launch, you can do "Edit->Delete memory" to make sure that you start from a clean memory. You may need to do this after starting the first bag with "--pause" so that rtabmap node is initialized to avoid a "service /reset cannot be called" error.

Launch: demo_multi-session_mapping.launch (https://github.com/introlab/rtabmap_ros/blob/master/launch/demo/demo_multi-session_mapping.launch)

```
$ roslaunch rtabmap_ros demo_multi-session_mapping.launch
$ rosbag play --clock --pause map1.bag
$ (...)
$ rosbag play --clock map2.bag
$ (...)
$ rosbag play --clock map3.bag
$ (...)
$ rosbag play --clock map4.bag
$ (...)
$ rosbag play --clock map5.bag
```

3.4 Robot mapping with Find-Object



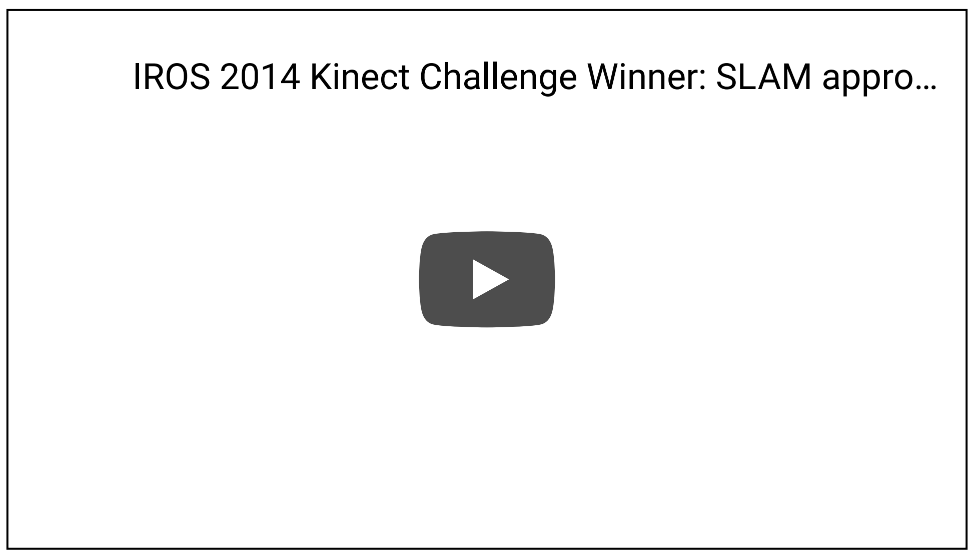
Find-Object's ros-pkg find_object_2d (/find_object_2d) should be installed.

ROS Bag: demo_find_object.bag (https://docs.google.com/uc?export=download&confirm=XJb5&id=0B46akLGdg-uaUEVEX1VmQTdOQUE) (416 MB)

Launch: demo_find_object.launch (https://github.com/introlab/rtabmap_ros/blob/master/launch/demo/demo_find_object.launch)

```
$ roslaunch rtabmap_ros demo_find_object.launch
$ rosbag play --clock demo_find_object.bag
```

3.5 IROS 2014 Kinect Challenge



There is no bag recorded for this demo but how to reproduce this setup is described on the page IROS 2014 Kinect Challenge (https://github.com/introlab/rtabmap/wiki/IROS-2014-Kinect-Challenge) of the RTAB-Map's wiki.

3.6 Stereo mapping

Visit the tutorial StereoOutdoorMapping (/rtabmap_ros/Tutorials/StereoOutdoorMapping) for detailed information. It is also shown how to create 2D occupancy grid map for navigation.

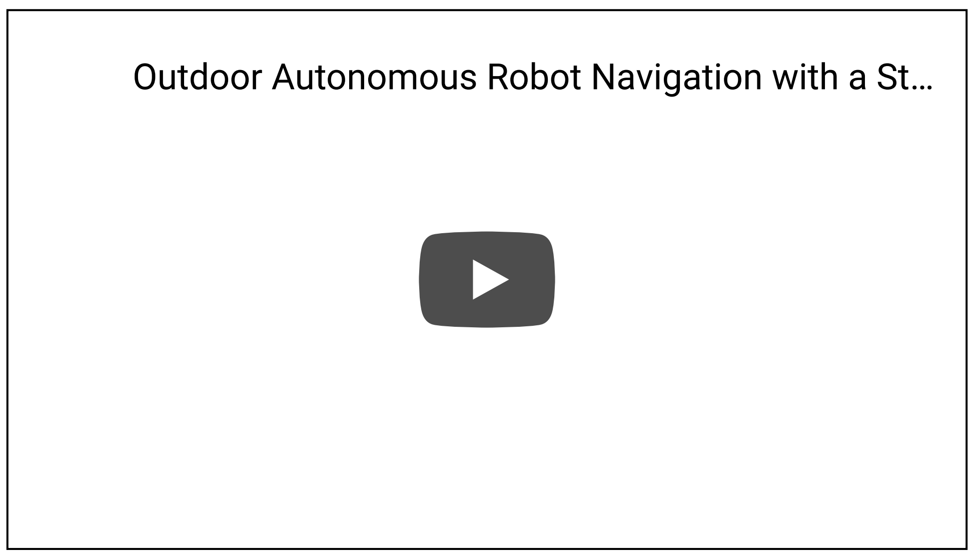
ROS bags:

- stereo_outdoorA.bag (https://docs.google.com/uc?export=download&confirm=ldd7&id=0B46akLGdg-uaOWtYT1ladldiS0U) (668 MB)
- stereo_outdoorB.bag (https://docs.google.com/uc?export=download&confirm=5Ptp&id=0B46akLGdg-uaVEs2SGFzT3ZVSU0) (987 MB)

Launch : demo_stereo_outdoor.launch (https://github.com/introlab/rtabmap_ros/blob/master/launch/demo/demo_stereo_outdoor.launch)

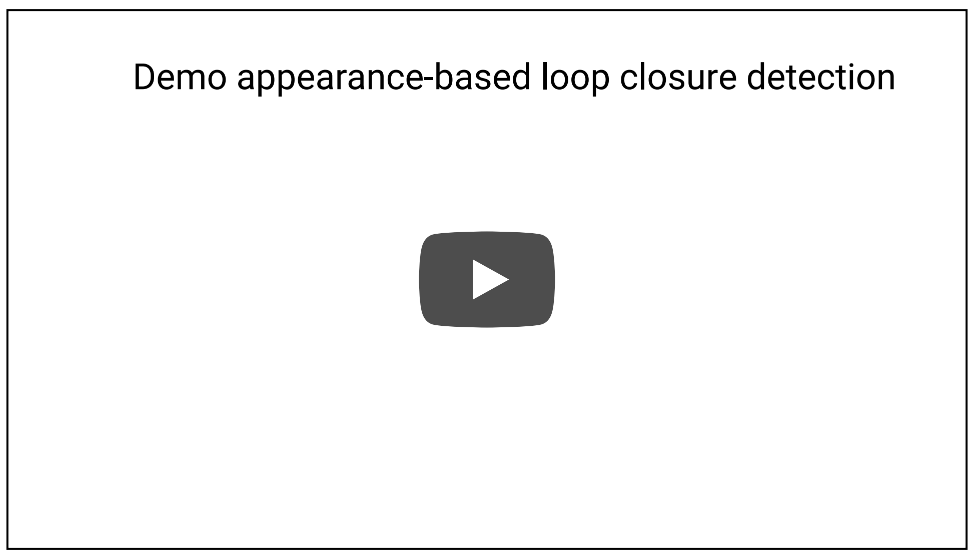
```
$ roslaunch rtabmap_ros demo_stereo_outdoor.launch
$ rosbag play --clock stereo_outdoorA.bag
[...]
$ rosbag play --clock stereo_outdoorB.bag
```

3.7 Stereo navigation



There is no bag recorded for this demo but how to reproduce this setup is described in the tutorial StereoOutdoorNavigation (/rtabmap_ros/Tutorials/StereoOutdoorNavigation).

3.8 Appearance-based loop closure detection-only



Data:

- samples.zip (https://github.com/introlab/rtabmap/raw/master/bin/data/samples.zip)
  - Set `video_or_images_path` parameter of camera (/rtabmap_ros#camera) node in the launch file below accordingly.

Launch: demo_appearance_mapping.launch (https://github.com/introlab/rtabmap_ros/blob/master/launch/demo/demo_appearance_mapping.launch)

```
$ roslaunch rtabmap_ros demo_appearance_mapping.launch
```

The GUI shows a plenty of information about the loop closures detected. If you only need the ID of the matched past image of a loop closure, you can do that:

```
$ rostopic echo /rtabmap/info/loopClosureId
6
---
0
---
7
---
```

A "0" means no loop closure detected. This can be also used in localization mode:

```
$ roslaunch rtabmap_ros demo_appearance_mapping.launch localization:=true
```

For **more videos** and information about the loop closure detection approach used in RTAB-Map, visit RTAB-Map on IntRoLab (https://introlab.3it.usherbrooke.ca/mediawiki-introlab/index.php?title=RTAB-Map&setlang=en).

4. Nodes

All sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html) topics use image_transport (/image_transport).

4.1 rtabmap

This is the main node of this package. It is a wrapper of the RTAB-Map Core library. This is where the graph of the map is incrementally built and optimized when a loop closure is detected. The online output of the node is the local graph with the latest added data to the map. The default location of the RTAB-Map database is "~/.ros/rtabmap.db" and the workspace is also set to "~/.ros". To get a 3D point cloud or a 2D occupancy grid of the environment, subscribe to `cloud_map`, `grid_map` or `proj_map` topics.

- There are two set of parameters: ROS and RTAB-Map's parameters. The ROS parameters are for connection stuff to interface the RTAB-Map library with ROS. The RTAB-Map's parameters are those from the RTAB-Map library.
    - One way to know RTAB-Map's parameters is to look at this file : Parameters.h (https://github.com/introlab/rtabmap/blob/master/corelib/include/rtabmap/core/Parameters.h#L161). There is a description for each parameter. Here an example (use string type for all RTAB-Map's parameters):

      ```
      <launch>
      <node name="rtabmap" pkg="rtabmap_ros" type="rtabmap">
         <param name="RGBD/OptimizeIterations" type="string" value="50"/>
      </node>
      </launch>
      ```

    - Another way to show available parameters from the terminal is to call the node with "--params" argument:

      ```
      $ rosrun rtabmap_ros rtabmap --params
      ```

    - By default, rtabmap is in mapping mode. To set in localization mode with a previously created map, you should set the memory not incremental (make sure that arguments don't contain `"--delete_db_on_start"` too!):

      ```
      <launch>
      <node name="rtabmap" pkg="rtabmap_ros" type="rtabmap" args="">
         <!-- LOCALIZATION MODE -->
         <param name="Mem/IncrementalMemory" type="string" value="false"/>
      </node>
      </launch>
      ```

4.1.1 Arguments

- `"--delete_db_on_start"` or `"-d"`: Delete the database before starting, otherwise the previous mapping session is loaded.
- `"--udebug"`: Show RTAB-Map's debug/info/warning/error logs.
- `"--uinfo"`: Show RTAB-Map's info/warning/error logs.
- `"--params"`: Show RTAB-Map's parameters related to this node and exit.

4.1.2 Subscribed Topics

`odom` (nav_msgs/Odometry (http://docs.ros.org/api/nav_msgs/html/msg/Odometry.html))
Odometry stream. Required if parameters `subscribe_depth` or `subscribe_stereo` are true **and** `odom_frame_id` is not set.

`rgb/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
RGB/Mono image. Should be rectified when `subscribe_depth` is true. Not required if parameter `subscribe_stereo` is true (use `left/image_rect` instead).

`rgb/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
RGB camera metadata. Not required if parameter `subscribe_stereo` is true (use `left/camera_info` instead).

`depth/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Registered depth image. Required if parameter `subscribe_depth` is true.

`scan` (sensor_msgs/LaserScan (http://docs.ros.org/api/sensor_msgs/html/msg/LaserScan.html))
Laser scan stream. Required if parameter `subscribe_scan` is true.

`scan_cloud` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
Laser scan point cloud stream. Required if parameter `subscribe_scan_cloud` is true.

`left/image_rect` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Left RGB/Mono rectified image. Required if parameter `subscribe_stereo` is true.

`left/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
Left camera metadata. Required if parameter `subscribe_stereo` is true.

`right/image_rect` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Right Mono rectified image. Required if parameter `subscribe_stereo` is true.

`right/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
Right camera metadata. Required if parameter `subscribe_stereo` is true.

`goal` (geometry_msgs/PoseStamped (http://docs.ros.org/api/geometry_msgs/html/msg/PoseStamped.html))
Planning Plan a path to reach this goal using the current online map. The goal should be close enough to the map's graph to be accepted (see RTAB-Map's parameter `RGBD/LocalRadius`). Note that only nodes in Working Memory are used, for exploration it may be ok, but it would be better to use service `set_goal` if you are coming back in a previously mapped area. See ***Planning*** published topics for corresponding generated outputs.

`rgbd_image` (rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html))
RGB-D synchronized image, only when `subscribe_rgbd` is `true`.

4.1.3 Published Topics

`info` (rtabmap_ros/Info (http://docs.ros.org/api/rtabmap_ros/html/msg/Info.html))
RTAB-Map's info.

`mapData` (rtabmap_ros/MapData (http://docs.ros.org/api/rtabmap_ros/html/msg/MapData.html))
RTAB-Map's graph and latest node data.

`mapGraph` (rtabmap_ros/MapGraph (http://docs.ros.org/api/rtabmap_ros/html/msg/MapGraph.html))
RTAB-Map's graph only.

`grid_map` (nav_msgs/OccupancyGrid (http://docs.ros.org/api/nav_msgs/html/msg/OccupancyGrid.html))
Mapping Occupancy grid generated with laser scans. Use parameters with prefixes `map_` and `grid_` below.

`proj_map` (nav_msgs/OccupancyGrid (http://docs.ros.org/api/nav_msgs/html/msg/OccupancyGrid.html))
Mapping Occupancy grid generated from projection of the 3D point clouds on the ground. Use parameters with prefixes `map_`, `grid_` and `proj_` below.

`cloud_map` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: ***Mapping*** 3D point cloud generated with the 3D point clouds. Use parameters with prefixes `map_` and `cloud_` below.

`scan_map` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: ***Mapping*** 3D point cloud generated with the 2D scans or 3D scans. Use parameters with prefixes `map_` and `scan_` below.

`labels` (visualization_msgs/MarkerArray (http://docs.ros.org/api/visualization_msgs/html/msg/MarkerArray.html))
: Convenient way to show graph's labels in RVIZ.

`global_path` (nav_msgs/Path (http://docs.ros.org/api/nav_msgs/html/msg/Path.html))
: ***Planning*** Poses of the planned global path. Published only once for each path planned.

`local_path` (nav_msgs/Path (http://docs.ros.org/api/nav_msgs/html/msg/Path.html))
: ***Planning*** Upcoming local poses corresponding to those of the global path. Published on every map update.

`goal_reached` (std_msgs/Bool (http://docs.ros.org/api/std_msgs/html/msg/Bool.html))
: ***Planning*** Status message if the goal is successfully reached or not.

`goal_out` (geometry_msgs/PoseStamped (http://docs.ros.org/api/geometry_msgs/html/msg/PoseStamped.html))
: ***Planning*** Current metric goal sent from the rtabmap's topological planner. For example, this can be connected to `move_base_simple/goal` topic of move_base (/move_base).

`octomap_full` (octomap_msgs/Octomap (http://docs.ros.org/api/octomap_msgs/html/msg/Octomap.html))
: Get an OctoMap (/OctoMap). *Available only if rtabmap_ros is built with octomap*.

`octomap_binary` (octomap_msgs/Octomap (http://docs.ros.org/api/octomap_msgs/html/msg/Octomap.html))
: Get an OctoMap (/OctoMap). *Available only if rtabmap_ros is built with octomap*.

`octomap_occupied_space` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: A point cloud of the occupied space (obstacles and ground) of the OctoMap (/OctoMap). *Available only if rtabmap_ros is built with octomap*.

`octomap_obstacles` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: A point cloud of the obstacles of the OctoMap (/OctoMap). *Available only if rtabmap_ros is built with octomap*.

`octomap_ground` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: A point cloud of the ground of the OctoMap (/OctoMap). *Available only if rtabmap_ros is built with octomap*.

`octomap_empty_space` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: A point cloud of empty space of the OctoMap (/OctoMap). *Available only if rtabmap_ros is built with octomap*.

`octomap_grid` (nav_msgs/OccupancyGrid (http://docs.ros.org/api/nav_msgs/html/msg/OccupancyGrid.html))
: The projection of the OctoMap (/OctoMap) into a 2D occupancy grid map. *Available only if rtabmap_ros is built with octomap*.

4.1.4 Services

`get_map` (rtabmap_ros/GetMap (http://docs.ros.org/api/rtabmap_ros/html/srv/GetMap.html))
: Call this service to get the map data

`get_grid_map` (nav_msgs/GetMap (http://docs.ros.org/api/nav_msgs/html/srv/GetMap.html))
: Call this service to get the occupancy grid built from laser scans

`get_proj_map` (nav_msgs/GetMap (http://docs.ros.org/api/nav_msgs/html/srv/GetMap.html))
: Call this service to get the occupancy grid built from the projection of the 3D point clouds

`publish_map` (rtabmap_ros/PublishMap (http://docs.ros.org/api/rtabmap_ros/html/srv/PublishMap.html))
: Call this service to publish the map data

`list_labels` (rtabmap_ros/ListLabels (http://docs.ros.org/api/rtabmap_ros/html/srv/ListLabels.html))
: Get current labels of the graph.

`update_parameters` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
: The node will update with current parameters of the rosparam server

`reset` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
: Delete the map

`pause` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
: Pause mapping

`resume` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
: Resume mapping

`trigger_new_map` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
: The node will begin a new map

`backup` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
: Backup the database to "`database_path`.back" (default `~/.ros/rtabmap.db.back`)

`set_mode_localization` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
: Set localization mode

`set_mode_mapping` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
: Set mapping mode

`set_label` (rtabmap_ros/SetLabel (http://docs.ros.org/api/rtabmap_ros/html/srv/SetLabel.html))
: Set a label to latest node or a specified node.

`set_goal` (rtabmap_ros/SetGoal (http://docs.ros.org/api/rtabmap_ros/html/srv/SetGoal.html))
: ***Planning*** Set a topological goal (a node id or a node label in the graph). Plan a path to reach this goal using the whole map contained in memory (including nodes in Long-Term Memory). See ***Planning*** published topics for corresponding generated outputs.

`octomap_full` (octomap_msgs/GetOctomap (http://docs.ros.org/api/octomap_msgs/html/srv/GetOctomap.html))
: Get an OctoMap (/OctoMap). *Available only if rtabmap_ros is built with octomap*.

`octomap_binary` (octomap_msgs/GetOctomap (http://docs.ros.org/api/octomap_msgs/html/srv/GetOctomap.html))
: Get an OctoMap (/OctoMap). *Available only if rtabmap_ros is built with octomap*.

4.1.5 Parameters

`~subscribe_depth` (`bool`, default: `"true"`)
: Subscribe to depth image

`~subscribe_scan` (`bool`, default: `"false"`)
: Subscribe to laser scan

`~subscribe_scan_cloud` (`bool`, default: `"false"`)
: Subscribe to laser scan point cloud

`~subscribe_stereo` (`bool`, default: `"false"`)
: Subscribe to stereo images

`~subscribe_rgbd` (`bool`, default: `"false"`)
: Subsribe to `rgbd_image` topic.

`~frame_id` (`string`, default: `"base_link"`)
: The frame attached to the mobile base.

`~map_frame_id` (`string`, default: `"map"`)
: The frame attached to the map.

`~odom_frame_id` (`string`, default: `""`)
: The frame attached to odometry. If empty, rtabmap will subscribe to `odom` topic to get odometry. If set, odometry is got from tf (/tf) (in this case, a covariance of 1 is used).

`~queue_size` (`int`, default: 10)
: Size of message queue for each synchronized topic.

`~publish_tf` (`bool`, default: `"true"`)
: Publish TF from /map to /odom.

`~tf_delay` (`double`, default: 0.05)
: Rate at which the TF from /map to /odom is published (20 Hz).

`~tf_prefix` (`string`, default: `""`)
: Prefix to add to generated tf (/tf).

`~wait_for_transform` (`bool`, default: `"true"`)
: Wait (maximum `wait_for_transform_duration` sec) for transform when a tf (/tf) transform is not still available.

`~wait_for_transform_duration` (`double`, default: 0.1)
: Wait duration for `wait_for_transform`.

`~config_path` (`string`, default: `""`)
: Path of a config files containing RTAB-Map's parameters. Parameters set in the launch file overwrite those in the config file.

`~database_path` (`string`, default: `"~/.ros/rtabmap.db"`)
: Path of the RTAB-Map's database.

`~gen_scan` (`bool`, default: `"false"`)
: Generate laser scans from depth images (using the middle horizontal line of the depth image). Not generated if `subscribe_scan` or `subscribe_scan_cloud` are true.

`~gen_scan_max_depth` (`double`, default: 4.0)
: Maximum depth of the laser scans generated.

`~approx_sync` (`bool`, default: `"false"`)
: Use approximate time synchronization of input messages. If false, note that the odometry input must have also exactly the same timestamps than the input images.

`~rgbd_cameras` (`int`, default: 1)
: Number of RGB-D cameras to use (when `subscribe_rgbd` is `true`). Well for now, a maximum of 4 cameras can be synchronized at the same time. If > 1, the rgbd_image topics should contain the camera index starting with 0. For example, if we have 2 cameras, you should set `rgbd_image0` and `rgbd_image1` topics.

`~use_action_for_goal` (`bool`, default: `"false"`)
: ***Planning*** Use actionlib (/actionlib) to send the metric goals to move_base (/move_base). The advantage over just connecting `goal_out` to `move_base_simple/goal` is that rtabmap can have a feedback if the goal is reached or if move_base (/move_base) has failed. See move_base Action API (/move_base#Action_API) for more info.

`~map_filter_radius` (`double`, default: 0.5)
: ***Mapping*** Filter nodes before creating the maps. Only load data for one node in the filter radius (the latest data is used) up to filter angle (`map_filter_angle`). Set to 0.0 to disable node filtering. Used for all published maps.

`~map_filter_angle` (`double`, default: 30.0)
: ***Mapping*** Angle used when filtering nodes before creating the maps. See also `map_filter_radius`. Used for all published maps.

`~map_cleanup` (`bool`, default: `"true"`)
: ***Mapping*** If there is no subscription to any map `cloud_map`, `grid_map` or `proj_map`, clear the corresponding data.

`~latch` (`bool`, default: `"true"`)
: ***Mapping*** If true, the last message published on the map topics will be saved and sent to new subscribers when they connect.

`~cloud_decimation` (`int`, default: 4)
: ***Mapping*** Image decimation before creating clouds added to cloud map. Set 1 to disable decimation. Used for `cloud_map` published topic.

`~cloud_max_depth` (`double`, default: 4.0)
: ***Mapping*** Maximum depth of the clouds added to cloud map. Set 0.0 to maximum depth filtering. Used for `cloud_map` published topic.

`~cloud_voxel_size` (`double`, default: 0.05)
: ***Mapping*** Voxel size used to filter each cloud added to cloud map. Set 0.0 to disable voxel filtering. Used for `cloud_map` published topic.

`~cloud_floor_culling_height` (`double`, default: 0.0)
: ***Mapping*** Filter the floor at the specified height (m). Used for `cloud_map` published topic.

`~cloud_output_voxelized` (`bool`, default: `"false"`)
: ***Mapping*** Do a final voxel filtering after all clouds are assembled. Used for `cloud_map` published topic.

`~cloud_frustum_culling` (`bool`, default: `"false"`)
: ***Mapping*** Filter the point cloud in the frustum defined by the last camera pose. Used for `cloud_map` published topic.

`~cloud_noise_filtering_radius` (`double`, default: 0)
: ***Mapping*** Filter points that have less neighbors than `cloud_noise_filtering_min_neighbors` in the radius `cloud_noise_filtering_radius` (0=disabled). Used for `cloud_map` published topic.

`~cloud_noise_filtering_min_neighbors` (`double`, default: 5)
: ***Mapping*** Filter points that have less neighbors than `cloud_noise_filtering_min_neighbors` in the radius `cloud_noise_filtering_radius` (0=disabled). Used for `cloud_map` published topic.

`~scan_voxel_size` (`double`, default: 0.05)
: ***Mapping*** Voxel size used to filter each scan added to scan map. Set 0.0 to disable voxel filtering. Used for `scan_map` published topic.

`~scan_output_voxelized` (`bool`, default: `"false"`)
: ***Mapping*** Do a final voxel filtering after all scans are assembled. Used for `scan_map` published topic.

`~grid_cell_size` (`double`, default: 0.05)
: ***Mapping*** Grid cell size (m). Used for `grid_map` and `proj_map` published topics.

`~grid_size` (`double`, default: 0)
: ***Mapping*** Initial size of the map (m). If 0, the map will grow as new data are added. If set, the map will still grow when the initial size is reached. Used for `grid_map` and `proj_map` published topics.

`~grid_eroded` (`bool`, default: `"false"`)
: ***Mapping*** Filter obstacle cells surrounded by empty space. Used for `grid_map` and `proj_map` published topics.

`~grid_unknown_space_filled` (`bool`, default: `"false"`)
: ***Mapping*** Fill empty space. Used for `grid_map` published topic.

`~proj_max_ground_angle` (`double`, default: 45)
: ***Mapping*** Maximum angle (degrees) between point's normal to ground's normal to label it as ground. Points with higher angle difference are considered as obstacles. Used for `proj_map` published topic.

`~proj_min_cluster_size` (`int`, default: 20)
: ***Mapping*** Minimum cluster size to project the points. The distance between clusters is defined by 2*`grid_cell_size`. Used for `proj_map` published topic.

`~proj_max_height` (`double`, default: 2.0)
: ***Mapping*** Maximum height of points used for projection. Used for `proj_map` published topic.

4.1.6 Required tf Transforms

`base_link` → `<the frame attached to sensors of incoming data>`
: usually a fixed value, broadcast periodically by a robot_state_publisher (/robot_state_publisher), or a `tf` static_transform_publisher (/tf#static_transform_publisher).

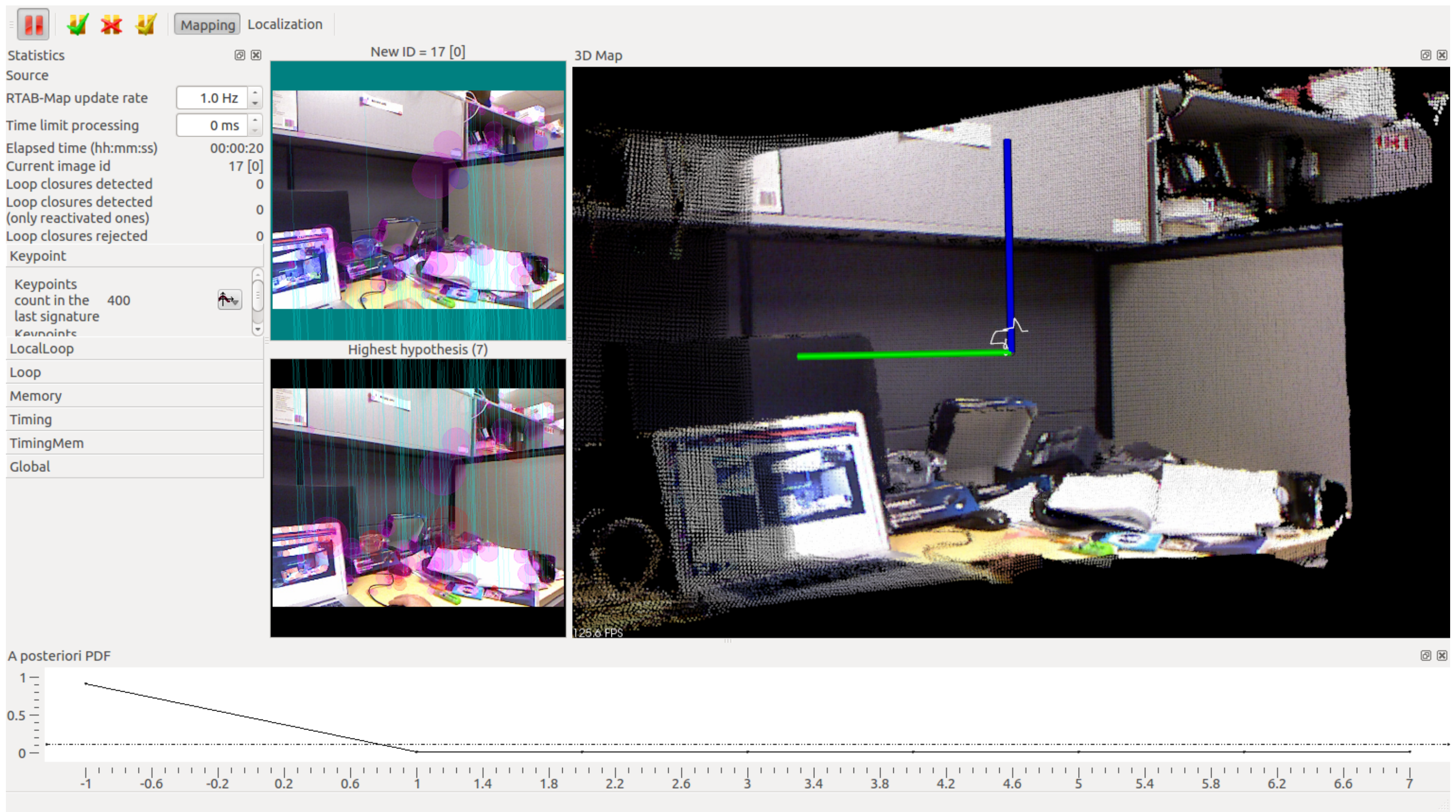
`odom` → `base_link`
: usually provided by the odometry system (e.g., the driver for the mobile base).

4.1.7 Provided tf Transforms

`map` → `odom`
: the current odometry correction.

4.2 rtabmapviz

This node starts the visualization interface of RTAB-Map. It is a wrapper of the RTAB-Map GUI library. It has the same purpose as rviz (/rviz) but with specific options for RTAB-Map.



RTAB-Map update rate 1.0 Hz
Time limit processing 0 ms
Elapsed time (hh:mm:ss) 00:00:20
Current image id 17 [0]
Loop closures detected 0
Loop closures detected (only reactivated ones) 0
Loop closures rejected 0
Keypoints count in the last signature 400

4.2.1 Arguments

- `-d "config.ini"`: Set a RTAB-Map ini file for GUI interface parameters. Default is `"/.ros/rtabmapGUI.ini"`.

4.2.2 Subscribed Topics

`odom` (nav_msgs/Odometry (http://docs.ros.org/api/nav_msgs/html/msg/Odometry.html))
: Odometry stream. Required if parameters `subscribe_depth` or `subscribe_stereo` are true **and** `odom_frame_id` is not set.

`rgb/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: RGB/Mono image. Should be rectified when `subscribe_depth` is true. Not required if parameter `subscribe_stereo` is true (use `left/image_rect` instead).

`rgb/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
: RGB camera metadata. Not required if parameter `subscribe_stereo` is true (use `left/camera_info` instead).

`depth/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: Registered depth image. Required if parameter `subscribe_depth` is true.

`scan` (sensor_msgs/LaserScan (http://docs.ros.org/api/sensor_msgs/html/msg/LaserScan.html))
: Laser scan stream. Required if parameter `subscribe_scan` is true.

`scan_cloud` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: Laser scan stream. Required if parameter `subscribe_scan_cloud` is true.

`left/image_rect` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: Left RGB/Mono rectified image. Required if parameter `subscribe_stereo` is true.

`left/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
: Left camera metadata. Required if parameter `subscribe_stereo` is true.

`right/image_rect` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: Right Mono rectified image. Required if parameter `subscribe_stereo` is true.

`right/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
: Right camera metadata. Required if parameter `subscribe_stereo` is true.

`odom_info` (rtabmap_ros/OdomInfo (http://docs.ros.org/api/rtabmap_ros/html/msg/OdomInfo.html))
: Odometry info. Required if parameter `subscribe_odom_info` is true.

`info` (rtabmap_ros/Info (http://docs.ros.org/api/rtabmap_ros/html/msg/Info.html))
: RTAB-Map's statistics info.

`mapData` (rtabmap_ros/MapData (http://docs.ros.org/api/rtabmap_ros/html/msg/MapData.html))
: RTAB-Map's graph and latest node data.

`rgbd_image` (rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html))
: RGB-D synchronized image, only when `subscribe_rgbd` is `true`.

4.2.3 Parameters

`~subscribe_depth` (`bool`, default: `"false"`)
: Subscribe to depth image

`~subscribe_scan` (`bool`, default: `"false"`)
: Subscribe to laser scan

`~subscribe_scan_cloud` (`bool`, default: `"false"`)
: Subscribe to laser scan point cloud

`~subscribe_stereo` (`bool`, default: `"false"`)
: Subscribe to stereo images

`~subscribe_odom_info` (`bool`, default: `"false"`)
: Subscribe to odometry info messages

`~subscribe_rgbd` (`bool`, default: `"false"`)
: Subsribe to `rgbd_image` topic.

`~frame_id` (`string`, default: `"base_link"`)
: The frame attached to the mobile base.

`~odom_frame_id` (`string`, default: `""`)
: The frame attached to odometry. If empty, rtabmapviz will subscribe to `odom` topic to get odometry. If set, odometry is got from tf (/tf).

`~tf_prefix` (`string`, default: `""`)
: Prefix to add to generated tf (/tf).

`~wait_for_transform` (`bool`, default: `"false"`)
: Wait (maximum 1 sec) for transform when a tf (/tf) transform is not still available.

`~queue_size` (`int`, default: 10)
: Size of message queue for each synchronized topic.

`~rgbd_cameras` (`int`, default: 1)
: Number of RGB-D cameras to use (when `subscribe_rgbd` is `true`). Well for now, a maximum of 4 cameras can be synchronized at the same time. If > 1, the rgbd_image topics should contain the camera index starting with 0. For example, if we have 2 cameras, you should set `rgbd_image0` and `rgbd_image1` topics.

4.2.4 Required tf Transforms

`base_link` → `<the frame attached to sensors of incoming data>`
: usually a fixed value, broadcast periodically by a robot_state_publisher (/robot_state_publisher), or a `tf` static_transform_publisher (/tf#static_transform_publisher).

`odom` → `base_link`
: usually provided by the odometry system (e.g., the driver for the mobile base).

`map` → `odom`
: the current odometry correction.

4.3 Visual and Lidar Odometry

Common odometry stuff for rgbd_odometry (/rtabmap_ros#rgbd_odometry), stereo_odometry (/rtabmap_ros#stereo_odometry) and icp_odometry (/rtabmap_ros#icp_odometry) nodes. These nodes wrap the various odometry approaches of RTAB-Map. When a transformation cannot be computed, a null transformation is sent to notify the receiver that odometry is not updated or lost.

- There are two set of parameters: ROS and RTAB-Map's parameters. The ROS parameters are for connection stuff to interface the RTAB-Map library with ROS. The RTAB-Map's parameters are those from the RTAB-Map library.
  - Parameters available for odometry can be shown from the terminal by using the "--params" argument:

```
$ rosrun rtabmap_ros rgbd_odometry --params
or
$ rosrun rtabmap_ros stereo_odometry --params
or
$ rosrun rtabmap_ros icp_odometry --params
```

4.3.1 Arguments

- `"--params"`: Show RTAB-Map's parameters related to this node and exit.

4.3.2 Published Topics

`odom` (nav_msgs/Odometry (http://docs.ros.org/api/nav_msgs/html/msg/Odometry.html))
Odometry stream. Send null odometry if cannot be computed.

`odom_info` (rtabmap_ros/OdomInfo (http://docs.ros.org/api/rtabmap_ros/html/msg/OdomInfo.html))
Odometry info stream. RTAB-Map's parameter `Odom/FillInfoData` should be true to fill features information (default is `true`).

`odom_last_frame` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
3D features of the last frame used to estimate the transformation.

`odom_local_map` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
Local map of 3D features used as reference to estimate the transformation. Only published if RTAB-Map's parameter `Odom/Strategy=0`.

4.3.3 Services

`reset_odom` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
Restart odometry to identity transformation.

`reset_odom_to_pose` (rtabmap_ros/ResetPose (http://docs.ros.org/api/rtabmap_ros/html/srv/ResetPose.html))
Restart odometry to specified transformation. Format: `"x y z roll pitch yaw"`.

`pause_odom` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
Pause odometry.

`resume_odom` (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
Resume odometry.

4.3.4 Parameters

`~frame_id` (`string`, default: `"base_link"`)
The frame attached to the mobile base.

`~odom_frame_id` (`string`, default: `"odom"`)
The odometry frame.

`~publish_tf` (`bool`, default: `"true"`)
Publish TF from /odom to /base_link.

`~tf_prefix` (`bool`, default: `""`)
Prefix to add to generated tf (/tf).

`~wait_for_transform` (`bool`, default: `"false"`)
Wait (maximum 1 sec) for transform when a tf (/tf) transform is not still available.

`~initial_pose` (`string`, default: `""`)
The initial pose of the odometry. Format: `"x y z roll pitch yaw"`.

`~queue_size` (`int`, default: 10)
Size of message queue for each synchronized topic.

`~publish_null_when_lost` (`bool`, default: true)
Odometry is published with null transform when lost.

`~ground_truth_frame_id` (`string`, default: `""`)
This can be used to have odometry initialized at the current ground truth pose.

`~ground_truth_base_frame_id` (`string`, default: `""`)
See `ground_truth_frame_id` description.

`~guess_frame_id` (`string`, default: `""`)
Use this frame as guess to compute odometry, otherwise odometry guess is done from a constant motion model. `tf` published by this node will be the correction of actual odometry. If `guess_frame_id` is `odom_combined`, `odom_frame_id` is `odom` and `frame_id` is `base_footprint`, `odom` -> `odom_combined` is published by this node to have a `tf` tree like this: `/odom -> /odom_combined -> /base_link`.

`~guess_min_translation` (`float`, default: `0.0` m)
When `guess_frame_id` is set, don't update odometry unless the guess moved at least as much as this value.

`~guess_min_rotation` (`float`, default: `0.0` rad)
When `guess_frame_id` is set, don't update odometry unless the guess rotated at least as much as this value.

`~config_path` (`string`, default: `""`)
A config (ini) file with RTAB-Map's parameters.

4.3.5 Required tf Transforms

`base_link` → `<the frame attached to sensors of incoming data>`
usually a fixed value, broadcast periodically by a robot_state_publisher (/robot_state_publisher), or a `tf` static_transform_publisher (/tf#static_transform_publisher).

4.3.6 Provided tf Transforms

`odom` → `base_link`
the current estimate of the robot's pose within the odometry frame.

4.4 rgbd_odometry

This node wraps the RGBD odometry approach of RTAB-Map. Using RGBD images, odometry is computed using visual features extracted from the RGB images with their depth information from the depth images. Using the feature correspondences between the images, a RANSAC approach computes the transformation between the consecutive images.

See also Visual Odometry (/rtabmap_ros#Visual_and_Lidar_Odometry) for common odometry stuff used by this node.

4.4.1 Subscribed Topics

`rgb/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
RGB/Mono rectified image.

`rgb/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
RGB camera metadata.

`depth/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Registered depth image.

`rgbd_image` (rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html))
RGB-D synchronized image, only when `subscribe_rgbd` is `true`.

4.4.2 Parameters

~approx_sync (bool, default: "true")
: Use approximate time synchronization of rgb and depth messages. If false, the rgb and depth images must have the same timestamps.

~rgbd_cameras (int, default: 1)
: Number of RGB-D cameras to use (when subscribe_rgbd is true). Well for now, a maximum of 4 cameras can be synchronized at the same time. If > 1, the rgbd_image topics should contain the camera index starting with 0. For example, if we have 2 cameras, you should set rgbd_image0 and rgbd_image1 topics.

~subscribe_rgbd (bool, default: "false")
: Subsribe to rgbd_image topic.

4.5 stereo_odometry

This node wraps the stereo odometry approach of RTAB-Map. Using stereo images, odometry is computed using visual features extracted from the left images with their depth information computed by finding the same features on the right images. Using the feature correspondences, a RANSAC approach computes the transformation between the consecutive left images.

See also Visual Odometry (/rtabmap_ros#Visual_and_Lidar_Odometry) for common odometry stuff used by this node.

4.5.1 Subscribed Topics

left/image_rect (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: Left RGB/Mono rectified image.

left/camera_info (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
: Left camera metadata.

right/image_rect (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: Right Mono rectified image.

right/camera_info (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
: Right camera metadata.

rgbd_image (rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html))
: Stereo synchronized image, only when subscribe_rgbd is true. This topic should contain left image in rgb field and right image in depth field (with corresponding camera info).

4.5.2 Parameters

~approx_sync (bool, default: "false")
: Use approximate time synchronization of stereo messages. If false, the left/right images and the left/right camera infos must have the same timestamps.

~subscribe_rgbd (bool, default: "false")
: Subsribe to rgbd_image topic.

4.6 icp_odometry

This node wraps the icp odometry approach of RTAB-Map. Using laser scan or a point cloud, odometry is computed by Iterative Closest Point (ICP) registration.

See also Visual Odometry (/rtabmap_ros#Visual_and_Lidar_Odometry) for common odometry stuff used by this node.

4.6.1 Subscribed Topics

scan (sensor_msgs/LaserScan (http://docs.ros.org/api/sensor_msgs/html/msg/LaserScan.html))
: Laser scan, assuming the lidar is installed horizontally. Don't use scan or scan_cloud at the same time.

scan_cloud (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: Point cloud, which can be 2D or 3D. Don't use scan or scan_cloud at the same time.

4.6.2 Parameters

~scan_cloud_max_points (int, default: 0)
: Maximum point in laser scan or point cloud. 0 means the maximum is defined by the size of ranges vector for laser scan topic.

~scan_downsampling_step (int, default: 1)
: Downsample the laser scan or point cloud. <=1 means no downsampling, otherwise one ray/point each step is kept.

~scan_voxel_size (float, default: 0.0 m)
: Filter the laser scan or point cloud using voxel filter (http://pointclouds.org/documentation/tutorials/voxel_grid.php) of this size. 0.0 means disabled.

~scan_normal_k (int, default: 0)
: Estimate normals of the laser scan or point cloud by using k nearest points. Useful only if Icp/Point2Plane parameter is enabled. 0 means disabled.

~scan_normal_radius (float, default: 0.0)
: Estimate normals of the laser scan or point cloud by using nearest points under this fixed radius. Useful only if Icp/Point2Plane parameter is enabled. 0.0 means disabled.

4.7 camera

A node for image acquisition from an USB camera (OpenCV is used). A special option for this node is that it can be configured to read images from a directory or a video file. Parameters can be changed with the dynamic_reconfigure (/dynamic_reconfigure) GUI from ROS. For dynamic parameters, see Camera.cfg (https://github.com/introlab/rtabmap_ros/blob/master/cfg/Camera.cfg)

4.7.1 Published Topics

image (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: Image stream.

4.7.2 Services

stop_camera (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
: Stop the camera.

start_camera (rtabmap_ros/ResetPose (http://docs.ros.org/api/rtabmap_ros/html/srv/ResetPose.html))
: Start the camera.

4.7.3 Parameters

~frame_id (string, default: "camera")
: The frame attached to the camera.

4.8 map_assembler

Note: It is recommend to use directly cloud_map or proj_map published topics from rtabmap (/rtabmap_ros#rtabmap) node instead of using this node.

This node subscribes to rtabmap (/rtabmap_ros#rtabmap) output topic "mapData" and assembles the 3D map incrementally, then publishes the same maps than rtabmap (/rtabmap_ros#rtabmap). See all ***Mapping*** related topics and parameters of rtabmap (/rtabmap_ros#rtabmap) node.

4.8.1 Subscribed Topics

mapData (rtabmap_ros/MapData (http://docs.ros.org/api/rtabmap_ros/html/msg/MapData.html))
: RTAB-Map's graph and latest node data.

4.8.2 Services

~reset (std_srvs/Empty (http://docs.ros.org/api/std_srvs/html/srv/Empty.html))
: Reset the cache.

4.9 map_optimizer

This node is for **advanced usage only** as it is preferred to use graph optimization already inside rtabmap (/rtabmap_ros#rtabmap) node (which is the default). See related parameters in rtabmap (/rtabmap_ros#rtabmap):

```
$rosrun rtabmap_ros rtabmap --params | grep Optimize
```

This node subscribes to rtabmap (/rtabmap_ros#rtabmap) output topic "mapData" and optimize the graph, then republishes the optimized "mapData".

- This node can be used to optimize the graph outside rtabmap (/rtabmap_ros#rtabmap) node. The benefice to do that is that we can keep optimized the global map instead of the local map of rtabmap (/rtabmap_ros#rtabmap). You can then connect output `mapData_optimized` to map_assembler (/rtabmap_ros#map_assembler) to get the optimized grid, proj and cloud maps assembled again. **Note that processing time for map optimization using this node is not bounded** (which is the case in rtabmap (/rtabmap_ros#rtabmap) node).
- You could also use your own graph optimization approach instead of this node by modifying `poses` values of the rtabmap_ros/MapData (http://docs.ros.org/api/rtabmap_ros/html/msg/MapData.html) msg. However, implementing your graph optimization approach inside rtabmap (/rtabmap_ros#rtabmap) is preferred (inherit Optimizer (https://github.com/introlab/rtabmap/blob/463a5d973c3bab96382356dd763a98caf9beec34/corelib/include/rtabmap/core/Graph.h#L44-L109) class and add it here (https://github.com/introlab/rtabmap/blob/463a5d973c3bab96382356dd763a98caf9beec34/corelib/src/Graph.cpp#L124-L137) with a new number, then you could select it after using the parameter `RGBD/OptimizeStrategy`).
- When using graph optimization outside rtabmap (/rtabmap_ros#rtabmap) node, you should set parameters `RGBD/OptimizeIterations` to `0`, `RGBD/OptimizeMaxError` to `0` and `publish_tf` to `false` for rtabmap (/rtabmap_ros#rtabmap) node.
- This node should not be used if rtabmap (/rtabmap_ros#rtabmap) node is in localization mode.

4.9.1 Subscribed Topics

`mapData` (rtabmap_ros/MapData (http://docs.ros.org/api/rtabmap_ros/html/msg/MapData.html))
RTAB-Map's graph and latest node data.

4.9.2 Published Topics

`[mapData]_optimized` (rtabmap_ros/MapData (http://docs.ros.org/api/rtabmap_ros/html/msg/MapData.html))
RTAB-Map's optimized graph and latest node data.

`[mapData]Graph_optimized` (rtabmap_ros/MapGraph (http://docs.ros.org/api/rtabmap_ros/html/msg/MapGraph.html))
RTAB-Map's optimized graph.

4.9.3 Parameters

`~map_frame_id` (`string`, default: `"map"`)
The frame attached to the map.

`~odom_frame_id` (`string`, default: `"odom"`)
The frame attached to odometry.

`~strategy` (`int`, default: 0)
Optimization approach used: 0=TORO, 1=g2o and 2=GTSAM.

`~slam_2d` (`bool`, default: `"false"`)
3DoF (x,y,yaw) optimization instead of 6DoF (x,y,z,roll,pitch,yaw).

`~robust` (`bool`, default: `"true"`)
Activate Vertigo robust graph optimization when g2o or GTSAM `strategy` is used.

`~global_optimization` (`bool`, default: `"true"`)
Optimize the global map. If false, only the local map is optimized.

`~iterations` (`int`, default: 100)
Map optimization iterations.

`~epsilon` (`double`, default: 0.0)
Stop graph optimization when error is less than `epsilon` (0=ignore epsilon).

`~ignore_variance` (`bool`, default: `"false"`)
Ignore constraints' variance. If checked, identity information matrix is used for each constraint in TORO. Otherwise, an information matrix is generated from the variance saved in the links.

`~optimize_from_last_node` (`bool`, default: `"false"`)
Optimize from the last node. If false, the graph is optimized from the oldest node linked to the current map.

`~publish_tf` (`bool`, default: `"true"`)
Publish TF from /map to /odom.

`~tf_delay` (`double`, default: 0.05)
Rate at which the TF from /map to /odom is published (20 Hz).

5. Nodelets

5.1 rtabmap_ros/rgbd_sync

Synchronize RGB, depth and camera_info messages into a single message. You can then use `subscribe_rgbd` to make `rtabmap` or odometry nodes subscribing to this message instead. This is useful when, for example, rtabmap is subscribed also to a laser scan or odometry topic published at different rate than the image topics. We can then make sure that images are correctly synchronized together. If you have camera publishing on the network, this can be also a good format to synchronize images before sending them on the network, to avoid synchronization issues when the network is lagging.

5.1.1 Subscribed Topics

`rgb/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
RGB image stream.

`depth/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Registered depth image stream.

`rgb/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
RGB camera metadata.

5.1.2 Published Topics

`rgbd_image` (rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html))
The RGB-D image topic

`rgbd_image/compressed` (rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html))
The compressed RGB-D image topic (rgb=jpeg, depth=png)

5.1.3 Parameters

`~queue_size` (`int`, default: 10)
Size of message queue for each synchronized topic.

`~approx_sync` (`bool`, default: `"True"`)
Use approximate synchronization for the input topics. If false, the RGB and depth images and the camera info must have the same timestamp.

`~compressed_rate` (`double`, default: 0)
Throttling rate at which `rgbd_image/compressed` topic will be published. 0 means no throttling.

5.2 rtabmap_ros/stereo_sync

Synchronize left image, right image and camera_info messages into a single message. You can then use `subscribe_rgbd` to make `rtabmap` or odometry nodes subscribing to this message instead. This is useful when, for example, rtabmap is subscribed also to a laser scan or odometry topic published at different rate than the image topics. We can then make sure that images are correctly synchronized together. If you have camera publishing on the network, this can be also a good format to synchronize images before sending them on the network, to avoid synchronization issues when the network is lagging.

5.2.1 Subscribed Topics

`left/image_rect` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Left image stream.

`right/image_rect` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Right image stream.

`left/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
Left camera metadata.

`right/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
Right camera metadata.

5.2.2 Published Topics

`rgbd_image` (rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html))
The RGB-D image topic. The RGB field is the left image and the depth field is the right image, with corresponding camera info.

`rgbd_image/compressed` (rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html))
The compressed RGB-D image topic (images compressed in jpeg). The RGB field is the left image and the depth field is the right image, with corresponding camera info.

5.2.3 Parameters

`~queue_size` (`int`, default: 10)
Size of message queue for each synchronized topic.

`~approx_sync` (`bool`, default: `"False"`)
Use approximate synchronization for the input topics. If false, the left image, right image and the camera info must have the same timestamp (which should be always the case for stereo images).

`~compressed_rate` (`double`, default: 0)
Throttling rate at which `rgbd_image/compressed` topic will be published. 0 means no throttling.

5.3 rtabmap_ros/rgbd_relay

A relay for rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html) messages. It works like a topic_tools/relay (http://wiki.ros.org/topic_tools/relay), but can also compress or uncompress data for convenience.

5.3.1 Subscribed Topics

`rgbd_image` (rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html))
RGB-D image input stream.

5.3.2 Published Topics

`[rgbd_image]_relay` (rtabmap_ros/RGBDImage (http://docs.ros.org/api/rtabmap_ros/html/msg/RGBDImage.html))
RGB-D image output stream.

5.3.3 Parameters

`~queue_size` (`int`, default: 10)
Size of message queue.

`~compress` (`bool`, default: `"False"`)
Publish compressed RGB-D image data.

`~uncompress` (`bool`, default: `"False"`)
Publish uncompressed RGB-D image data.

5.4 rtabmap_ros/data_odom_sync

Synchronize odometry with RGB-D images. Useful to correctly show clouds in RVIZ when odometry refresh rate is low comparatively to clouds to show.

5.4.1 Subscribed Topics

`odom_in` (nav_msgs/Odometry (http://docs.ros.org/api/nav_msgs/html/msg/Odometry.html))
Odometry stream.

`rgb/image_in` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
RGB image stream.

`depth/image_in` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Registered depth image stream.

`rgb/camera_info_in` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
RGB camera metadata.

5.4.2 Published Topics

`odom_out` (nav_msgs/Odometry (http://docs.ros.org/api/nav_msgs/html/msg/Odometry.html))
Odometry stream.

`rgb/image_out` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
RGB image stream.

`depth/image_out` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Registered depth image stream.

`rgb/camera_info_out` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
RGB camera metadata.

5.4.3 Parameters

`~queue_size` (`int`, default: 10)
Size of message queue for each synchronized topic.

5.5 rtabmap_ros/data_throttle

Throttle at a specified rate the OpenNI data. See also rtabmap_ros/rgbd_sync (/rtabmap_ros#rgbd_sync), which is now preferred when publishing remotely.

5.5.1 Subscribed Topics

`rgb/image_in` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
RGB image stream.

`depth/image_in` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Registered depth image stream.

`rgb/camera_info_in` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
RGB camera metadata.

5.5.2 Published Topics

`rgb/image_out` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
RGB image stream.

`depth/image_out` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
Registered depth image stream.

`rgb/camera_info_out` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
RGB camera metadata.

5.5.3 Parameters

`~queue_size` (`int`, default: 10)
Size of message queue for each synchronized topic.

`~rate` (`double`, default: 0)
Maximum frame rate (Hz).

`~approx_sync` (`bool`, default: `"True"`)
Use approximate synchronization for the input topics. If false, the RGB and depth images and the camera info must have the same timestamp.

`~decimation` (`double`, default: 1)
Optional decimation of the RGB and depth images (with corresponding modifications in `camera_info_out`).

5.6 rtabmap_ros/stereo_throttle

Throttle at a specified rate the stereo data.

5.6.1 Subscribed Topics

`left/image_rect` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
    Left RGB/Mono rectified image.

`left/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
    Left camera metadata.

`right/image_rect` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
    Right Mono rectified image.

`right/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
    Right camera metadata.

5.6.2 Published Topics

`[left/image_rect]_throttle` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
    Left RGB/Mono rectified image.

`[left/camera_info]_throttle` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
    Left camera metadata.

`[right/image_rect]_throttle` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
    Right Mono rectified image.

`[right/camera_info]_throttle` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
    Right camera metadata.

5.6.3 Parameters

`~queue_size` (`int`, default: 10)
    Size of message queue for each synchronized topic.

`~rate` (`double`, default: 0)
    Maximum frame rate (Hz).

`~approx_sync` (`bool`, default: `"false"`)
    Use approximate time synchronization of stereo messages. If false, the left/right images and the left/right camera infos must have the same timestamp.

`~decimation` (`double`, default: 1)
    Optional decimation of the left and right images (with corresponding modifications in camera infos).

5.7 rtabmap_ros/point_cloud_xyzrgb

Construct a point cloud from RGB and depth images or stereo images. Optional decimation, depth, voxel and noise filtering can be applied.

5.7.1 Subscribed Topics

`rgb/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
    RGB image stream.

`depth/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
    Registered depth image stream.

`rgb/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
    RGB camera metadata.

`left/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
    Left RGB/Mono rectified image.

`left/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
    Left camera metadata.

`right/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
    Right Mono rectified image.

`right/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
    Right camera metadata.

5.7.2 Published Topics

`cloud` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
    Generated RGB point cloud.

5.7.3 Parameters

`~queue_size` (`int`, default: 10)
    Size of message queue for each synchronized topic.

`~approx_sync` (`bool`, default: `"true"`)
    Use approximate time synchronization of the input topics. If false, the input topics must have the same timestamp (set to `"false"` for stereo images).

`~decimation` (`int`, default: 1)
    Decimation of the images before creating the point cloud. Set to 1 to not decimate the images.

`~voxel_size` (`double`, default: 0.0)
    Voxel size (m) of the generated cloud. Set `0.0` to deactivate voxel filtering.

`~min_depth` (`double`, default: 0.0)
    Min depth (m) of the generated cloud.

`~max_depth` (`double`, default: 0.0)
    Max depth (m) of the generated cloud. Set `0.0` to deactivate depth filtering.

`~noise_filter_radius` (`double`, default: 0.0)
    Max radius (m) for searching point neighbors. Set `0.0` to deactivate noise filtering.

`~noise_filter_min_neighbors` (`int`, default: 5)
    Minimum neighbors of a point to keep it.

5.8 rtabmap_ros/point_cloud_xyz

Construct a point cloud from a depth or disparity image. Optional decimation, depth, voxel and noise filtering can be applied.

5.8.1 Subscribed Topics

`depth/image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
    Depth image stream.

`depth/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
    Depth camera metadata.

`disparity/image` (stereo_msgs/DisparityImage (http://docs.ros.org/api/stereo_msgs/html/msg/DisparityImage.html))
    Disparity image stream.

`disparity/camera_info` (sensor_msgs/CameraInfo (http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html))
    Disparity camera metadata.

5.8.2 Published Topics

`cloud` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
    Generated point cloud.

5.8.3 Parameters

`~queue_size` (`int`, default: 10)
    Size of message queue for each synchronized topic.

`~approx_sync` (`bool`, default: `"true"`)
: Use approximate time synchronization of messages. If false, the image and camera info must have the same timestamps.

`~decimation` (`int`, default: 1)
: Decimation of the RGB/depth images before creating the point cloud. Set to 1 to not decimate the images.

`~voxel_size` (`double`, default: 0.0)
: Voxel size (m) of the generated cloud. Set `0.0` to deactivate voxel filtering.

`~min_depth` (`double`, default: 0.0)
: Min depth (m) of the generated cloud.

`~max_depth` (`double`, default: 0.0)
: Max depth (m) of the generated cloud. Set `0.0` to deactivate depth filtering.

`~noise_filter_radius` (`double`, default: 0.0)
: Max radius (m) for searching point neighbors. Set `0.0` to deactivate noise filtering.

`~noise_filter_min_neighbors` (`int`, default: 5)
: Minimum neighbors of a point to keep it.

5.9 rtabmap_ros/disparity_to_depth

Convert a disparity image to a depth image.

5.9.1 Subscribed Topics

`disparity` (stereo_msgs/DisparityImage (http://docs.ros.org/api/stereo_msgs/html/msg/DisparityImage.html))
: Disparity image stream.

5.9.2 Published Topics

`depth` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: Depth image stream in m. Format TYPE_32FC1.

`depth_raw` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: Depth image stream in mm. Format TYPE_16UC1.

5.10 rtabmap_ros/pointcloud_to_depthimage

Reproject a point cloud into a camera frame to create a depth image. When `fixed_frame_id` is set (e.g., "odom"), movement is taken into account by transforming the point cloud accordingly to camera timestamp. An example of usage of this nodelet can be found in the Tango ROS Streamer (http://wiki.ros.org/rtabmap_ros/Tutorials/Tango%20ROS%20Streamer) tutorial.

5.10.1 Subscribed Topics

`camera_info` (stereo_msgs/CameraInfo (http://docs.ros.org/api/stereo_msgs/html/msg/CameraInfo.html))
: Camera info in which we want to reproject the points.

`cloud` (stereo_msgs/PointCloud2 (http://docs.ros.org/api/stereo_msgs/html/msg/PointCloud2.html))
: The cloud to reproject in the camera.

5.10.2 Published Topics

`image` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: Depth image stream in m. Format TYPE_32FC1.

`image_raw` (sensor_msgs/Image (http://docs.ros.org/api/sensor_msgs/html/msg/Image.html))
: Depth image stream in mm. Format TYPE_16UC1.

5.10.3 Parameters

`~queue_size` (`int`, default: 10)
: Size of message queue for each synchronized topic.

`~fixed_frame_id` (`string`, default: "")
: This frame should be set when using approximate time synchronization (`approx` is `true`). If the robot is moving, it could be "odom". If not moving, it could be "base_link".

`~approx` (`bool`, default: `"true"`)
: Approximate time synchronization.

`~wait_for_transform` (`double`, default: 0.1)
: Wait duration for transform when a tf (/tf) transform is not still available.

`~decimation` (`int`, default: 1)
: Scale down image size of the camera info received (creating smaller depth image).

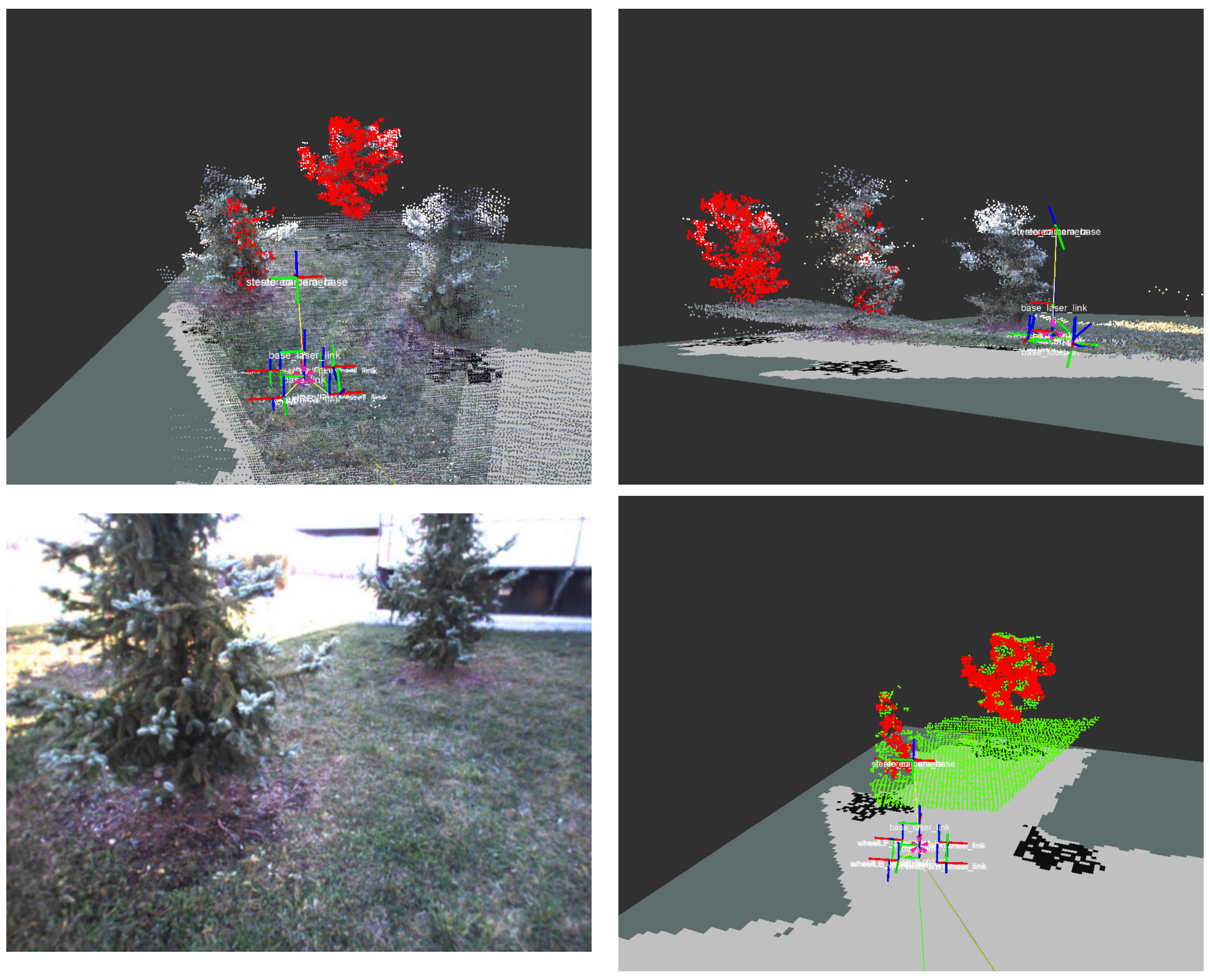
`~fill_holes_size` (`int`, default: 0)
: Fill holes of empty pixels up to this size. Values are interpolated from neighbor depth values. 0 means disabled.

`~fill_holes_error` (`double`, default: 0.1)
: Maximum depth error (m) to interpolate.

`~fill_iterations` (`int`, default: 1)
: Number of iterations to fill holes.

5.11 rtabmap_ros/obstacles_detection

This nodelet extracts obstacles and the ground from a point cloud. The camera must see the ground to work reliably. Since the ground is not even, the ground is segmented by normal filtering: all points with normal in the +z direction (+- fixed angle) are labelled as ground, all the others are labelled as obstacles. The images below show an example.



5.11.1 Subscribed Topics

`cloud` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: A point cloud generated from a RGB-D camera or a stereo camera.

5.11.2 Published Topics

`ground` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: The segmented ground.

`obstacles` (sensor_msgs/PointCloud2 (http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html))
: The segmented obstacles.

5.11.3 Parameters

`~frame_id` (`string`, default: `"base_link"`)
: The frame attached to the mobile base.

`~queue_size` (`int`, default: 10)
: Size of message queue for each synchronized topic.

`~normal_estimation_radius` (`double`, default: 0.05)
: Normal estimation radius (m).

`~ground_normal_angle` (`double`, default: PI/4)
: Maximum angle from the +z axis of the point's normal to be labelled as ground.

`~min_cluster_size` (`int`, default: 20)
: Minimum size of the segmented clusters to keep.

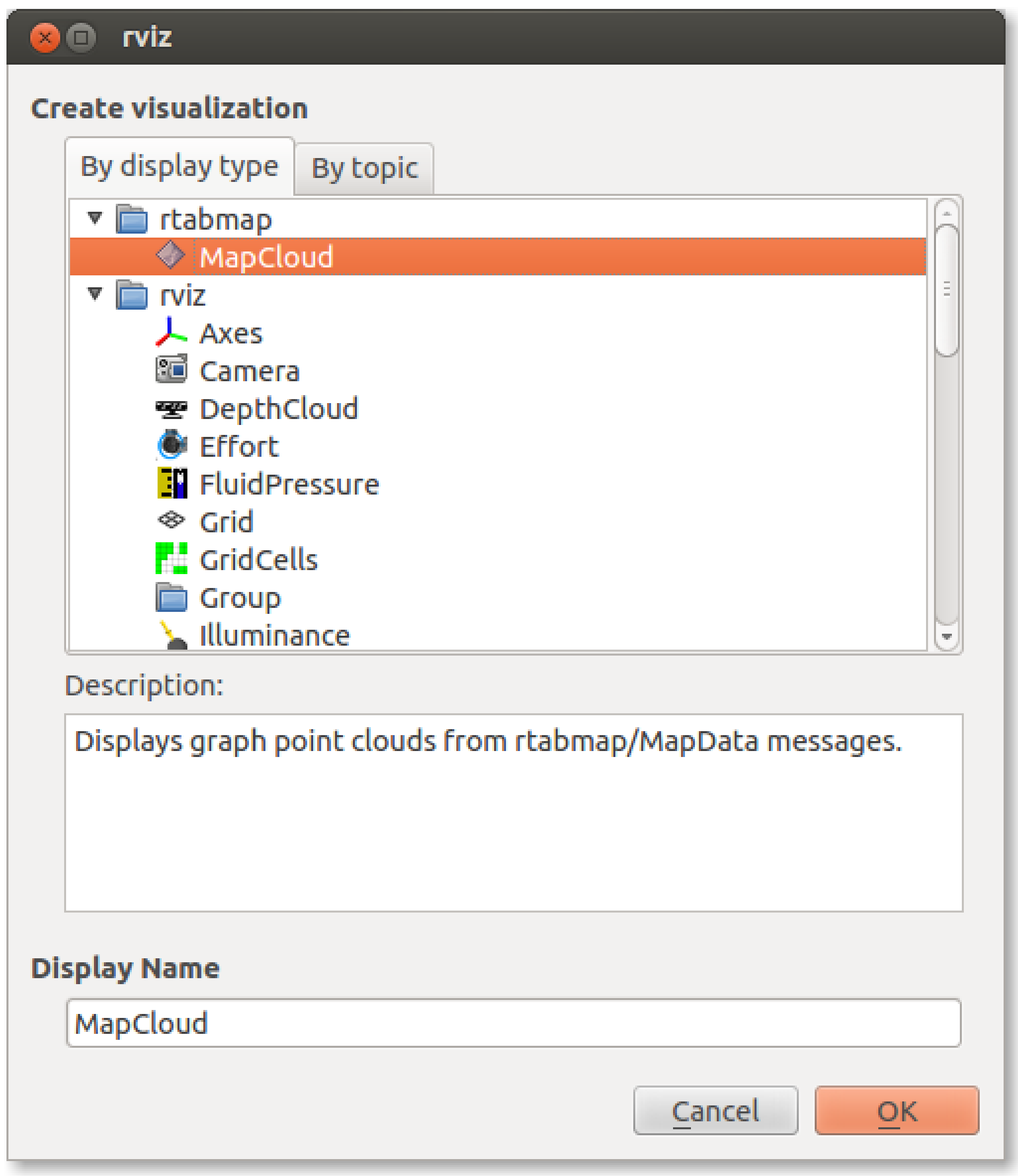
`~max_obstacles_height` (`double`, default: 0.0)
: Maximum height of obstacles.

5.11.4 Required tf Transforms

`base_link` → `<the frame attached to sensors of incoming data>`
: usually a fixed value, broadcast periodically by a robot_state_publisher (/robot_state_publisher), or a `tf` static_transform_publisher (/tf#static_transform_publisher).

6. RVIZ plugins

6.1 Map Cloud Display



Description:

Displays graph point clouds from rtabmap/MapData messages.

Display Name

MapCloud

This rviz plugin subscribes to /mapData (rtabmap_ros/MapData (http://docs.ros.org/api/rtabmap_ros/html/msg/MapData.html)) topic. A 3D map cloud will be created incrementally in RVIZ. When the graph is changed, all point clouds added in RVIZ will be transformed to new poses. It has the same properties as PointCloud (/rviz/DisplayTypes/PointCloud) display but with these new ones:

Properties

Name	Description	Valid Values	Default
Cloud decimation	How the input depth and RGB images are decimated before creating the point cloud	[1-16]	4
Cloud max depth (m)	Maximum depth of each point cloud added to map (0 means no maximum)	[0-9999]	4
Cloud voxel size (m)	Voxel size of the generated point clouds (0 means no voxel)	[0-1]	0.01
Filter floor (m)	Maximum height of the floor filtered (disabled=0).	[0-9999]	0
Node filtering radius (m)	Only keep one node in the specified radius (disabled=0).	[0-1]	0.2
Node filtering angle (degrees)	Only keep one node in the specified angle in the filtering radius (disabled=0).	[0-180]	30
Download Map	Download the map from rtabmap (/rtabmap_ros#rtabmap) node. This may take a while if the map is big, be patient!	.	.
Download Graph	Download the graph from rtabmap (/rtabmap_ros#rtabmap) node.	.	.

6.2 Map Graph Display

This rviz plugin subscribes to /mapGraph (rtabmap_ros/MapGraph (http://docs.ros.org/api/rtabmap_ros/html/msg/MapGraph.html)) topic. It will show the RTAB-Map's graph with different colors depending on the links' type. It has the same properties as Path (/rviz/DisplayTypes/Path) display.

6.3 Info Display

This rviz plugin subscribes to /info (rtabmap_ros/Info (http://docs.ros.org/api/rtabmap_ros/html/msg/Info.html)) topic. Information about loop closures detected are shown in the "Status".

Wiki: rtabmap_ros (last edited 2019-04-04 18:08:07 by MathieuLabbe (/MathieuLabbe))

Please ask about problems and questions regarding this tutorial on answers.ros.org (http://answers.ros.org). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

Advanced Parameter Tuning

Description: This tutorial tells you which parameter to change to improve performances

Keywords: parameters

Tutorial Level: INTERMEDIATE

Contents

1. Introduction

This tutorial is aimed at helping people using rtabmap in a more advanced way. Only after you have successfully installed rtabmap and rtabmap_ros, you shall start this tutorial.

2. Change Parameters

To know what parameters you can change, and what do they mean, you can use rtabmap gui and check the Preferences dialog or you can do:

```
rosrun rtabmap_ros rtabmap --params
rosrun rtabmap_ros rgbd_odometry --params
```

If you want to know a specific parameter, for example about g2o

```
$ rosrun rtabmap_ros rtabmap --params | grep g2o
Param: Optimizer/Robust = "false" [Robust graph optimization using Vertigo (only work for g2o and GTSAM optimization strategies). Not compatible with "RGBD/OptimizeMaxError" if enabled.]
Param: Optimizer/Strategy = "2"   [Graph optimization strategy: 0=TORO, 1=g2o and 2=GTSAM.]
Param: g2o/Optimizer = "0"        [0=Levenberg 1=GaussNewton]
Param: g2o/PixelVariance = "1"    [Pixel variance used for SBA.]
Param: g2o/Solver = "0"           [0=csparse 1=pcg 2=cholmod]
```

There are generally three way of changing parameters.

1) Use a config.ini file.

The default config.ini should be automaticlly generated once you launch `rtabmap` through terminal by "`$ rtabmap`" command. Its location is `~/Documents/RTAB-Map/config.ini`. You can copy this `config.ini` file into your own package and change your launch file `cfg` parameter into the corresponding location.

For example, you can set `Odom\Strategy=1` in `config.ini`, then in a package called `arti_vision`, you can put the `config.ini` in the config folder and renamed it into `rtabmap.ini`. Calling `rtabmap.launch` with the `cfg` argument:

```
<arg name="rtabmap_args" default="$(find arti_vision)/config/rtabmap.ini" />
```

2) Use <param> tag

This is the normal ROS way of reading parameters. For exmaple, you can put this under `rgbd_odometry` node

```
<param name="Odom/Strategy" value="0"/>
```

3) Add parameters in arguments of the node.

If you just want to temporarily play with some parameters, you can quickly put it under the `rtabmap_args` argument:

```
<arg name="rtabmap_args" default="--Odom/Strategy 0"/>
```

All three way of changing parameters have the same effects. Choose whatever you want, though the second approach is the standard way to set parameters in ROS.

3. Visual Odometry

Tuning visual odometry parameter is very important, it directly affects how good RTAB-Map generally performs.

3.1 Increase Speed

If you have a very limited computing power, you probably want to play with the following parameters, so that you can increase the odometry frequency, and thus, the robot won't get lost. Here are some parameters you should try:

3.1.1 Change the Visual Odometry Stategy into Frame to Frame

```
<!-- 0=Frame-to-Map (F2M) 1=Frame-to-Frame (F2F) -->
<param name="Odom/Strategy" value="1"/>
```

3.1.2 Change the Matching Correspondences into Optical Flow

Optical flow may give more matches, but less robust correspondences:

```
<!-- Correspondences: 0=Features Matching, 1=Optical Flow -->
<param name="Vis/CorType" value="1"/>
```

3.1.3 Reduce the Maximum Features

```
<!-- maximum features map size, default 2000 -->
<param name="OdomF2M/MaxSize" type="string" value="1000"/>
<!-- maximum features extracted by image, default 1000 -->
<param name="Vis/MaxFeatures" type="string" value="600"/>
```

3.1.4 Increase the Frame Rate of Camera and Reduce the Resolution

Thoes two asspects are very effective, increase your camera's frame rate directly affects how fast the VO can track your robot. Reduce resolution can greatly increase the speed of processing but may reduce the accuracy. You need to make a good balance based on your hardware and use case.

3.1.5 Change Feature Distance

If your images are 720p or more, you may want to increase GFTT distance (minimum distance between extracted features) so that you have more distributed features in the images:

```
<node pkg="rtabmap_ros" type="rgbd_odometry" name="rgbd_odometry">
        <param name="GFTT/MinDistance" type="string" value="10"/> <!-- default 5 pixels -->
</node>
```

3.2 Odometry Auto-Reset

When calling reset_odom service, rtabmap will start a new map, hiding the old one until a loop closure is found with the previous map. If you don't want rtabmap to start a new map when odometry is reset and wait until a first loop closure is found, you can set `Rtabmap/StartNewMapOnLoopClosure` to `true`. If parameter `Odom/ResetCountdown` is set to `1` (default `0=disabled`), odometry will automatically reset one frame after being lost, i.e., it has the same effect than calling `reset_odom` service.

```
<node pkg="rtabmap_ros" type="rgbd_odometry" name="rgbd_odometry">
        <param name="Odom/ResetCountdown"              value="1" />
<node/>
<node pkg="rtabmap_ros" type="rtabmap" name="rtabmap">
        <param name="Rtabmap/StartNewMapOnLoopClosure" value="true"/>
<node/>
```

4. Mapping

4.1 Three DOF mapping

For most of UGV, the vehicle only runs on a flat ground, in this way, you can force the visual odometry to track the vehicle in only 3DOF (x,y,theta) and increase the robustness of the map. For rtabmap, we can also constraint to 3 DoF loop closure detection and graph optimization:

```
<node pkg="rtabmap_ros" type="rgbd_odometry" name="rgbd_odometry">
        <param name="Reg/Force3DoF"    value="true" />
<node/>

<node pkg="rtabmap_ros" type="rtabmap" name="rtabmap">
        <param name="Reg/Force3DoF"    value="true" />
        <param name="Optimizer/Slam2D" value="true" />
<node/>
```

4.2 Change inital costmap size for move_base

You can initialize the minimum size of the map with `grid_size` parameter:

```
<node pkg="rtabmap_ros" type="rtabmap" name="rtabmap">
   <param name="grid_size" type="double" value="50"/> <!-- 50 meters wide -->
</node>
```

4.3 Reduce Point Cloud Noises

```
<param name="cloud_noise_filtering_radius" value="0.05"/>
<param name="cloud_noise_filtering_min_neighbors" value="2"/>
```

Reference: pcl::RadiusOutlierRemoval() (http://docs.pointclouds.org/1.7.0/classpcl_1_1_radius_outlier_removal.html)

4.4 Change Projected Occupancy Grid Characteristic

`proj_max_ground_angle` means mapping maximum angle between point's normal to ground's normal to label it as ground. Points with higher angle difference are considered as obstacles.

Increasing `proj_max_ground_angle` will make the algorithm include points with normal's angle farther from z+ axis as ground.

```
<param name="proj_max_ground_angle" value="45"/>
```

`proj_max_ground_height` means mapping maximum height of points used for projection. Used for `proj_map` published topic.

Increasing the `proj_max_ground_height` will make the algorithm ignore points that are under this threshold while projection. All points below this threshold will be ignored:

```
<param name="proj_max_ground_height" value="0.1"/>
```

`proj_max_height` means mapping maximum height of points used for projection:

```
<param name="proj_max_height" value="2.0"/>
```

`proj_min_cluster_size` means mapping minimum cluster size to project the points. The distance between clusters is defined by `2*grid_cell_size`:

```
<param name="proj_min_cluster_size" value="20"/>
```

Wiki: rtabmap_ros/Tutorials/Advanced Parameter Tuning (last edited 2017-03-29 15:11:31 by MathieuLabbe (/MathieuLabbe))