Institut de Robòtica i Informàtica Industrial

Leg Detector and MHT Tracker tutorial: Capabilities

Ely Repiso, Fernando Herrero, Alberto Sanfeliu.

European Commission | Horizon 2020 European Union funding for Research & Innovation

AI4EU TERRINet The European Robotics Research Infrastructure Network

This work has received partial founding from European Union's Horizon 2020 research and innovation program under the projects AI4EU (H2020-ICT-2018-2-825619) and TERRINet (H2020-INFRAIA-2017-1-two-stage-730994)

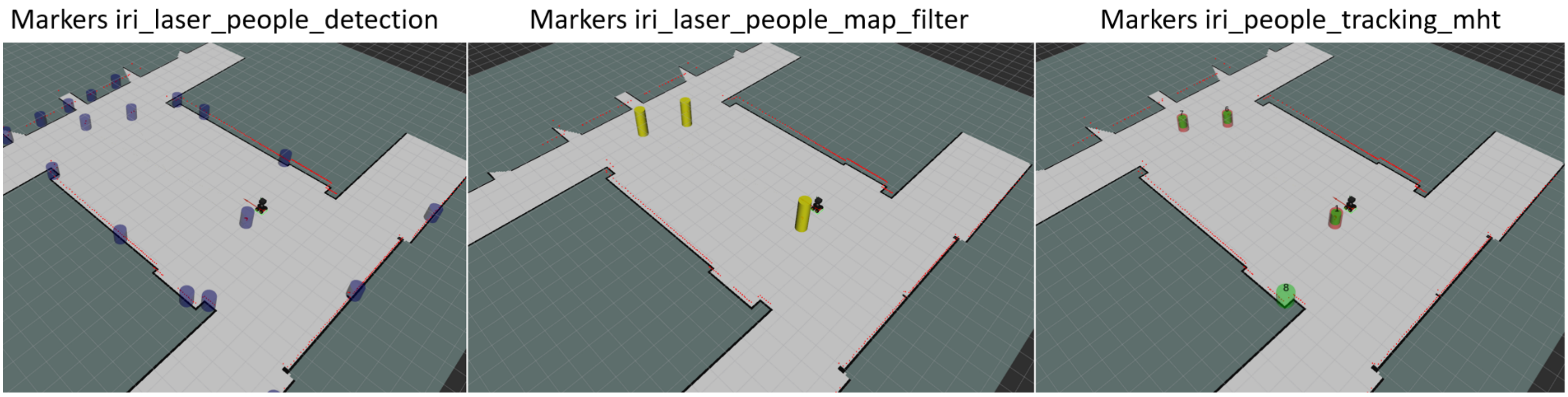
CSIC Consejo Superior de Investigaciones Científicas

Universitat Politècnica de Catalunya BarcelonaTech

In this document, we show the capabilites of the detector and tracker to be adapted to any environment and type of 2D laser.

1. Detect and track several people

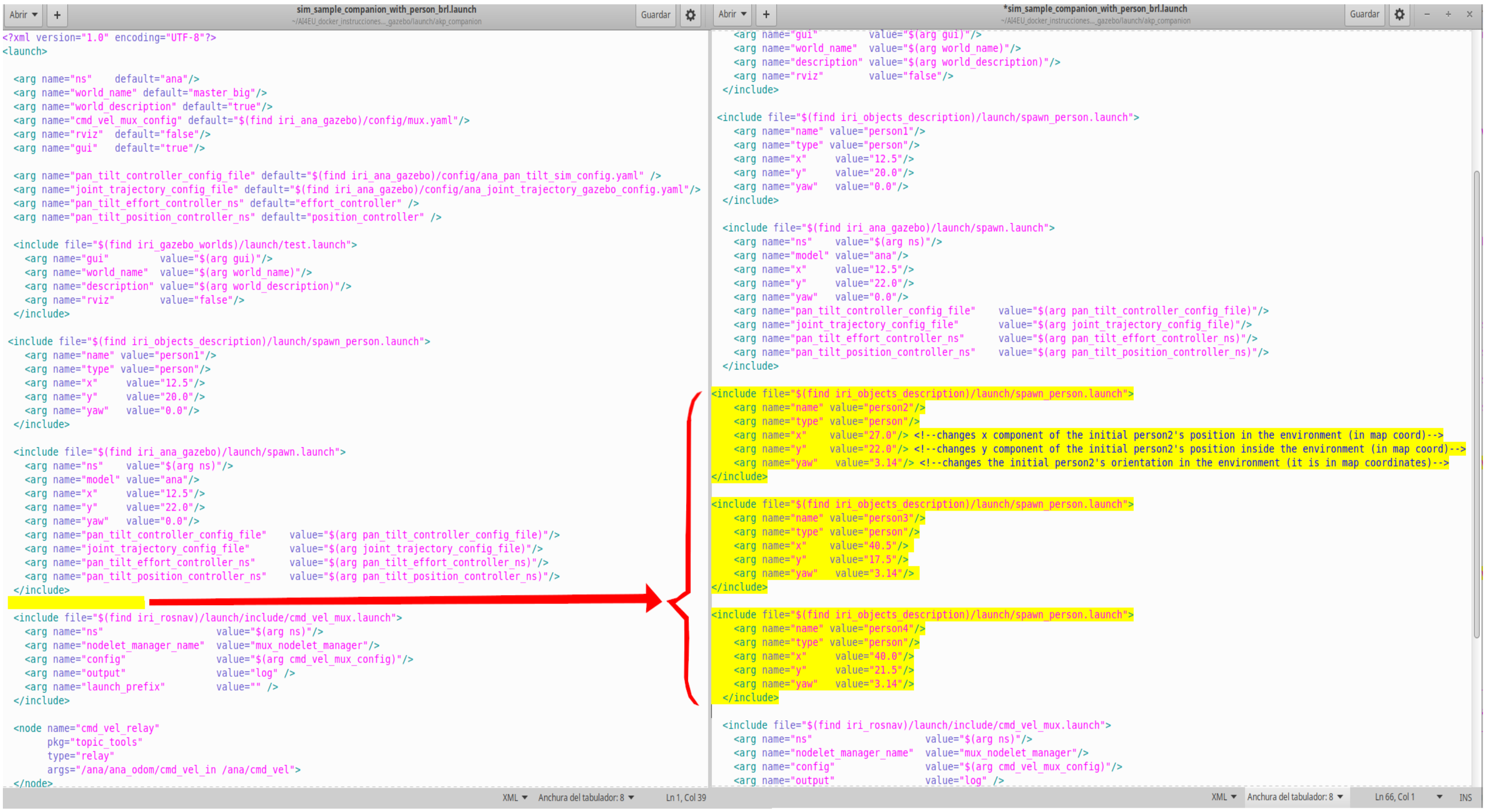
This two capabilities are from three ROS-nodes iri_laser_people_detection, iri_laser_people_map_filter (to filter the detections with the map) and iri_people_tracking_mht, which allow us to detect people by detecting the two semi-circles of the two legs seen by the 2D laser, filter these people detentions with any map of the environment and allow us to track people represented by the laser detection. These laser detection's are 2D points of the people position over the floor of any environment and the tracker is a multi-hypothesis tracker of detections represented by any 2D points in any 2D space. Next, we include an image that shows the outputs of each of these nodes represented by markers in the rviz. Also, we output detection and track messages with these ones to be used by other nodes.



2. Add more people in the environment

You can add other people in the environment to cross the group’s path and see the dynamic people avoidance of the robot’s accompaniment.

In order to include these people in the environment, you need to add them in the gazebo-launch. For example in: roslaunch iri_ana_gazebo sim_sample_companion_with_person_brl.launch world_name:=master_big. Then, you need to open the launch file sim_sample_companion_with_person_brl.launch and include more people, like the next image shows:



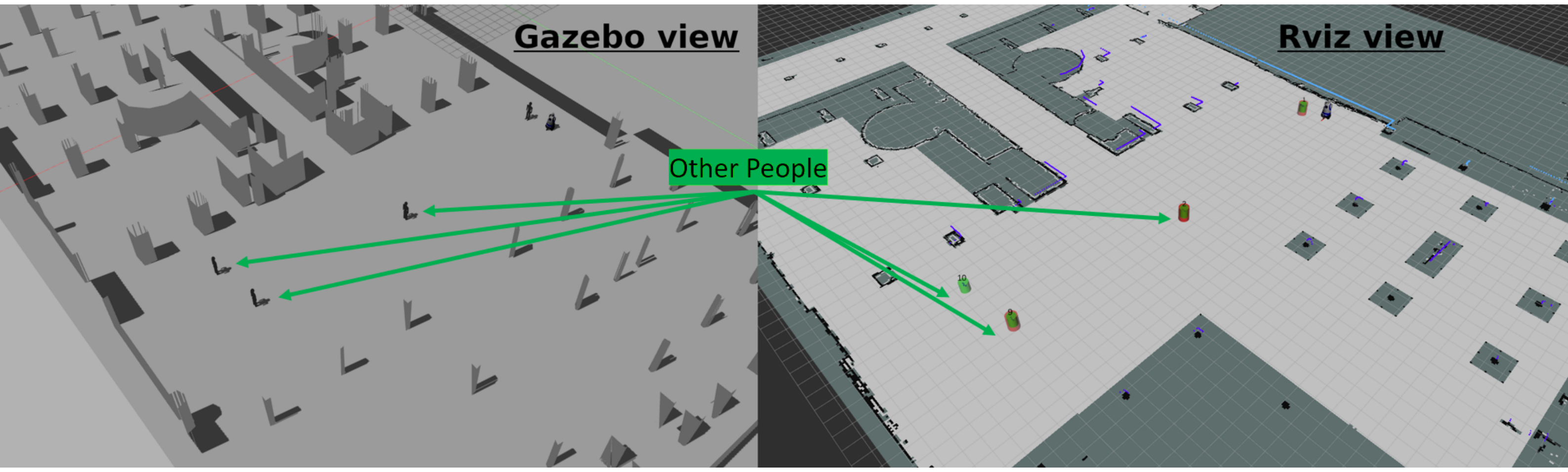
sim_sample_companion_with_person_brl.launch

```xml
<?xml version="1.0" encoding="UTF-8"?>
<launch>

  <arg name="ns"    default="ana"/>
  <arg name="world_name" default="master_big"/>
  <arg name="world_description" default="true"/>
  <arg name="cmd_vel_mux_config" default="$(find iri_ana_gazebo)/config/mux.yaml"/>
  <arg name="rviz"  default="false"/>
  <arg name="gui"   default="true"/>

  <arg name="pan_tilt_controller_config_file" default="$(find iri_ana_gazebo)/config/ana_pan_tilt_sim_config.yaml" />
  <arg name="joint_trajectory_config_file" default="$(find iri_ana_gazebo)/config/ana_joint_trajectory_gazebo_config.yaml"/>
  <arg name="pan_tilt_effort_controller_ns" default="effort_controller" />
  <arg name="pan_tilt_position_controller_ns" default="position_controller" />

  <include file="$(find iri_gazebo_worlds)/launch/test.launch">
    <arg name="gui"         value="$(arg gui)"/>
    <arg name="world_name"  value="$(arg world_name)"/>
    <arg name="description" value="$(arg world_description)"/>
    <arg name="rviz"        value="false"/>
  </include>

 <include file="$(find iri_objects_description)/launch/spawn_person.launch">
    <arg name="name" value="person1"/>
    <arg name="type" value="person"/>
    <arg name="x"     value="12.5"/>
    <arg name="y"     value="20.0"/>
    <arg name="yaw"   value="0.0"/>
  </include>

  <include file="$(find iri_ana_gazebo)/launch/spawn.launch">
    <arg name="ns"    value="$(arg ns)"/>
    <arg name="model" value="ana"/>
    <arg name="x"     value="12.5"/>
    <arg name="y"     value="22.0"/>
    <arg name="yaw"   value="0.0"/>
    <arg name="pan_tilt_controller_config_file"    value="$(arg pan_tilt_controller_config_file)"/>
    <arg name="joint_trajectory_config_file"       value="$(arg joint_trajectory_config_file)"/>
    <arg name="pan_tilt_effort_controller_ns"      value="$(arg pan_tilt_effort_controller_ns)"/>
    <arg name="pan_tilt_position_controller_ns"    value="$(arg pan_tilt_position_controller_ns)"/>
  </include>

  <include file="$(find iri_rosnav)/launch/include/cmd_vel_mux.launch">
    <arg name="ns"                   value="$(arg ns)"/>
    <arg name="nodelet_manager_name" value="mux_nodelet_manager"/>
    <arg name="config"               value="$(arg cmd_vel_mux_config)"/>
    <arg name="output"               value="log" />
    <arg name="launch_prefix"        value="" />
  </include>

  <node name="cmd_vel_relay"
        pkg="topic_tools"
        type="relay"
        args="/ana/ana_odom/cmd_vel_in /ana/cmd_vel">
  </node>
```

*sim_sample_companion_with_person_brl.launch

```xml
    <arg name="gui"         value="$(arg gui)"/>
    <arg name="world_name"  value="$(arg world_name)"/>
    <arg name="description" value="$(arg world_description)"/>
    <arg name="rviz"        value="false"/>
  </include>

 <include file="$(find iri_objects_description)/launch/spawn_person.launch">
    <arg name="name" value="person1"/>
    <arg name="type" value="person"/>
    <arg name="x"     value="12.5"/>
    <arg name="y"     value="20.0"/>
    <arg name="yaw"   value="0.0"/>
  </include>

  <include file="$(find iri_ana_gazebo)/launch/spawn.launch">
    <arg name="ns"    value="$(arg ns)"/>
    <arg name="model" value="ana"/>
    <arg name="x"     value="12.5"/>
    <arg name="y"     value="22.0"/>
    <arg name="yaw"   value="0.0"/>
    <arg name="pan_tilt_controller_config_file"    value="$(arg pan_tilt_controller_config_file)"/>
    <arg name="joint_trajectory_config_file"       value="$(arg joint_trajectory_config_file)"/>
    <arg name="pan_tilt_effort_controller_ns"      value="$(arg pan_tilt_effort_controller_ns)"/>
    <arg name="pan_tilt_position_controller_ns"    value="$(arg pan_tilt_position_controller_ns)"/>
  </include>

<include file="$(find iri_objects_description)/launch/spawn_person.launch">
    <arg name="name" value="person2"/>
    <arg name="type" value="person"/>
    <arg name="x"    value="27.0"/> <!--changes x component of the initial person2's position in the environment (in map coord)-->
    <arg name="y"    value="22.0"/> <!--changes y component of the initial person2's position inside the environment (in map coord)-->
    <arg name="yaw"   value="3.14"/> <!--changes the initial person2's orientation in the environment (it is in map coordinates)-->
</include>

<include file="$(find iri_objects_description)/launch/spawn_person.launch">
    <arg name="name" value="person3"/>
    <arg name="type" value="person"/>
    <arg name="x"    value="40.5"/>
    <arg name="y"    value="17.5"/>
    <arg name="yaw"   value="3.14"/>
</include>

<include file="$(find iri_objects_description)/launch/spawn_person.launch">
    <arg name="name" value="person4"/>
    <arg name="type" value="person"/>
    <arg name="x"     value="40.0"/>
    <arg name="y"     value="21.5"/>
    <arg name="yaw"   value="3.14"/>
  </include>

  <include file="$(find iri_rosnav)/launch/include/cmd_vel_mux.launch">
    <arg name="ns"                   value="$(arg ns)"/>
    <arg name="nodelet_manager_name" value="mux_nodelet_manager"/>
    <arg name="config"               value="$(arg cmd_vel_mux_config)"/>
    <arg name="output"               value="log" />
```

Now, if you include the same people in the launchs of Dabo-robot and execute the launch that includes Gazebo and the rviz (roslaunch iri_dabo_gazebo sim_gazebo_dabo_companion.launch), you see the result of the next image:



If you want to move these people, you have to use the next teleop-rosrun-commands inside a Ubuntu-terminal:

$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py cmd_vel:=/**person2**/cmd_vel __name:=**person2**

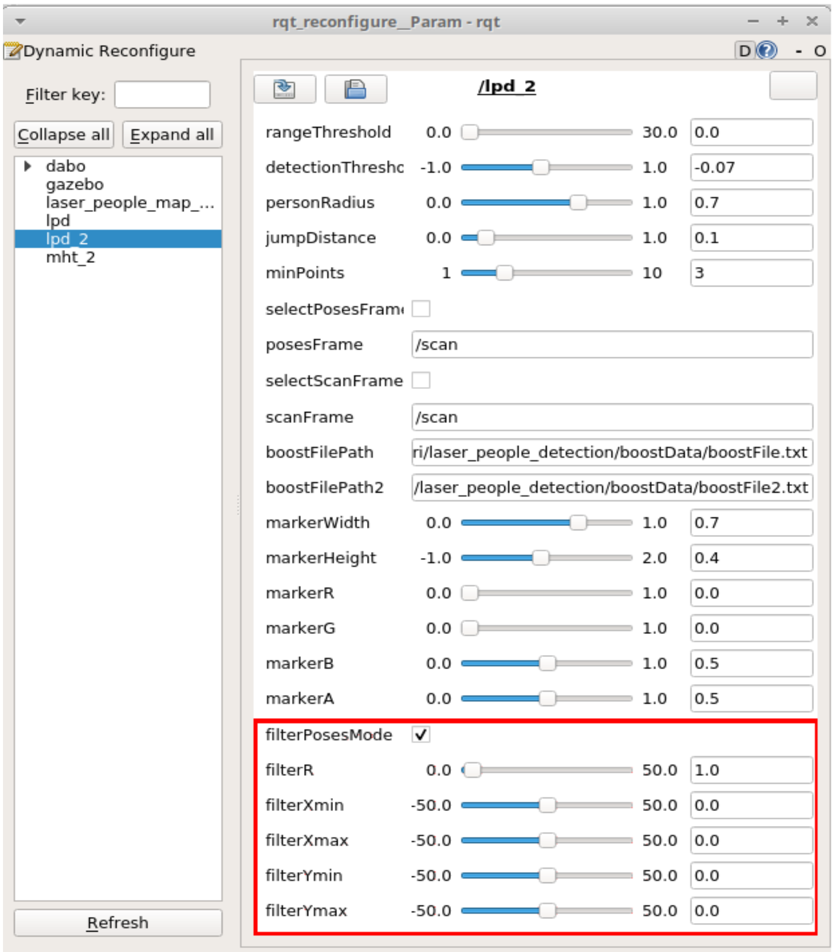
$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py cmd_vel:=/**personX**/cmd_vel __name:=**personX**

Where personX is the name of the person included, which needs to match with the name included in the launch to add this person.

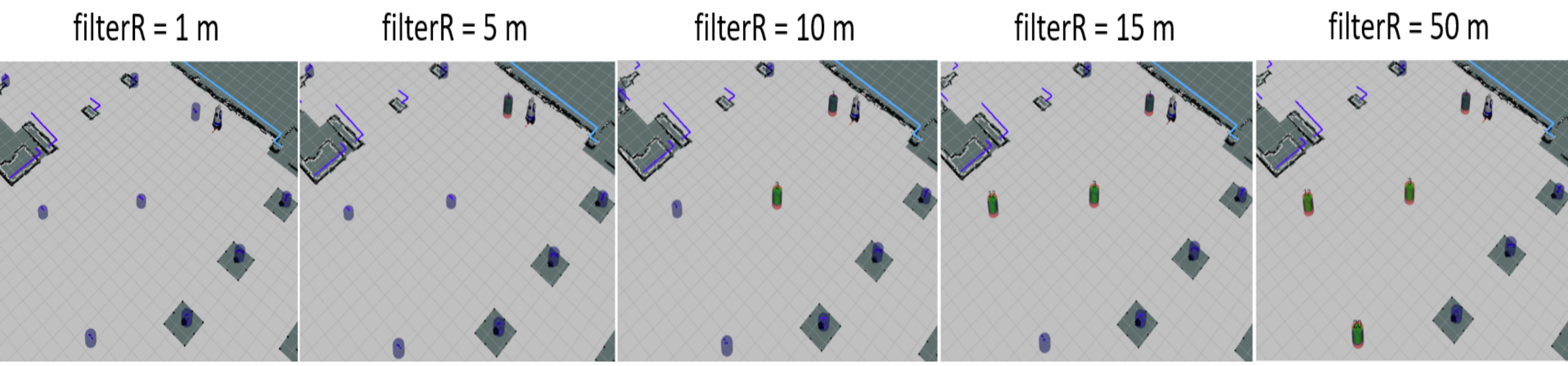
3. How to reduce the radii to detect people

You can reduce the radii around the robot to detect people. This reduction allows a minimization of the computational load of the algorithm.

<u>11.2 Reduce the radii to detect other people:</u> The reduction of this radii is better to be reduced directly in the laser-leg-detector, because you reduce at the same time the computational load of the tracker and the planner to do the accompaniment. In the rqt_reconfigure you need to arrive at lpd_2 or lpd (laser people detector). The parameters that you need to change are first the Boolean to allow the filtering, filterPosesMode=true, and second include which radii you want to filter around the robot by changing the value of filterR. Also, you may filter using different distance in X and in Y by using the parameters filterXmax and filterYmax, respectively.



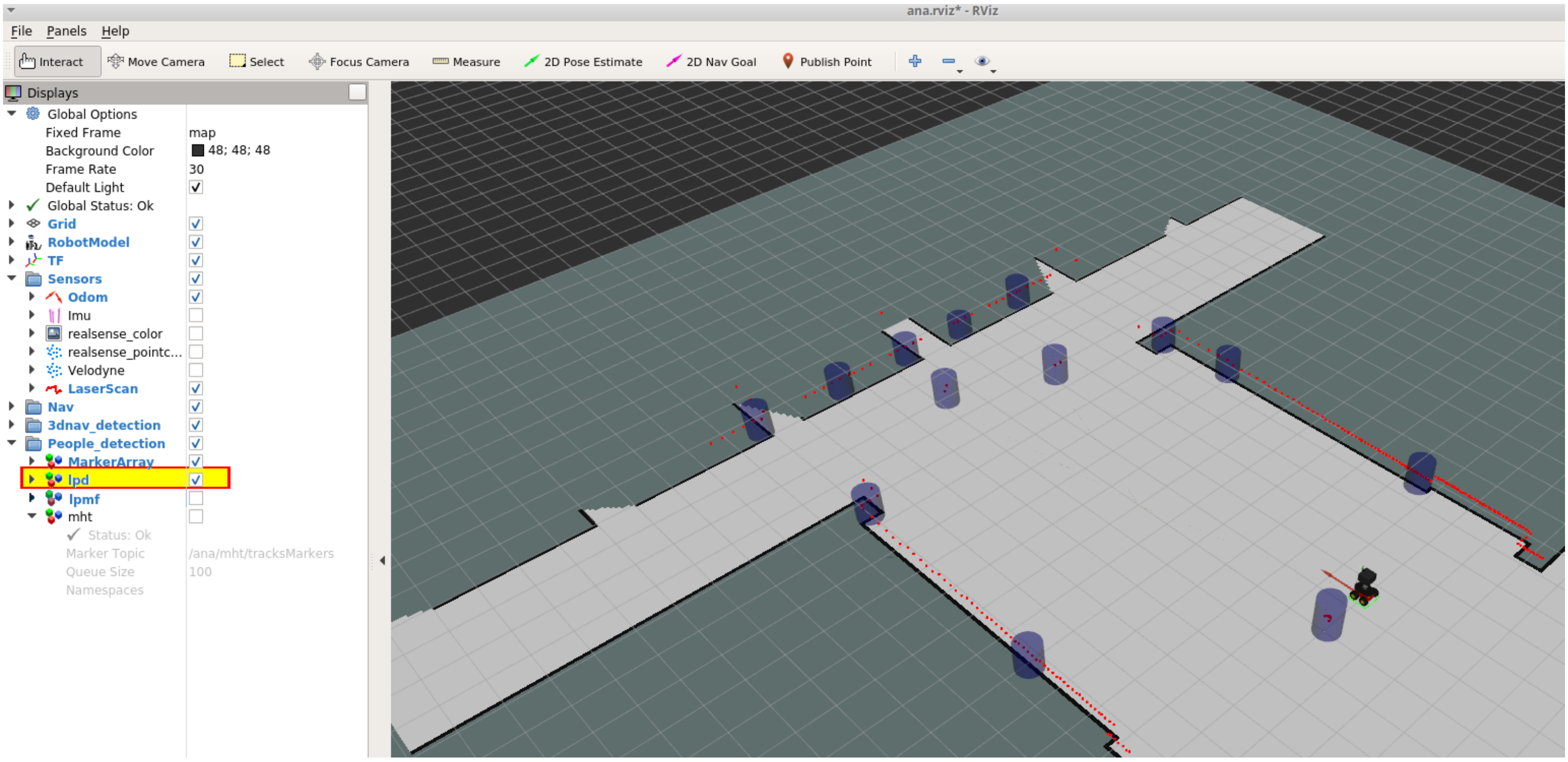
Next images show several examples of how affect the value of the parameter filterR in the distance around the robot position to track people and use these tracks in the planner to do the accompaniment. This distance is in meters. In the image the people that uses the robot to do the plan are the only ones that have track with identifier associated (in red and green with an id over them). The blue cylinders are all people detection, which are filtered at the output of the node using the value of the parameter filterR. With filterR=50 m you do not filter any thing. With filterR=1 m you filter all the people (also the accompanied one). The most recommended values are filterR=5 m, filterR=10 m or filterR=15 m, to choose one or other depend on the computational power of your system.



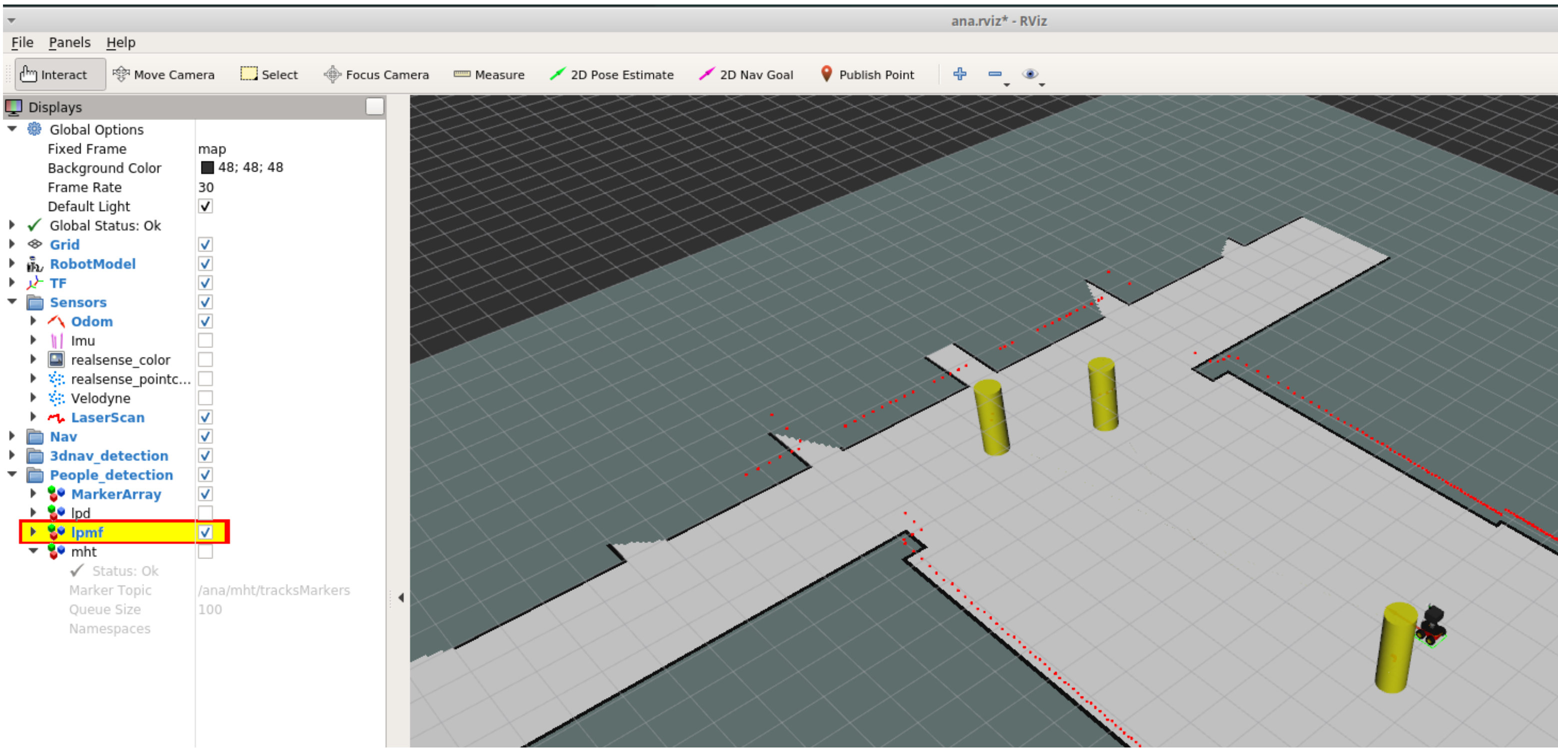
4. How to reduce the number of markers in rviz

You can reduce the number of markers to be seeing in the rviz. This reduction allows to obtain less computational load.

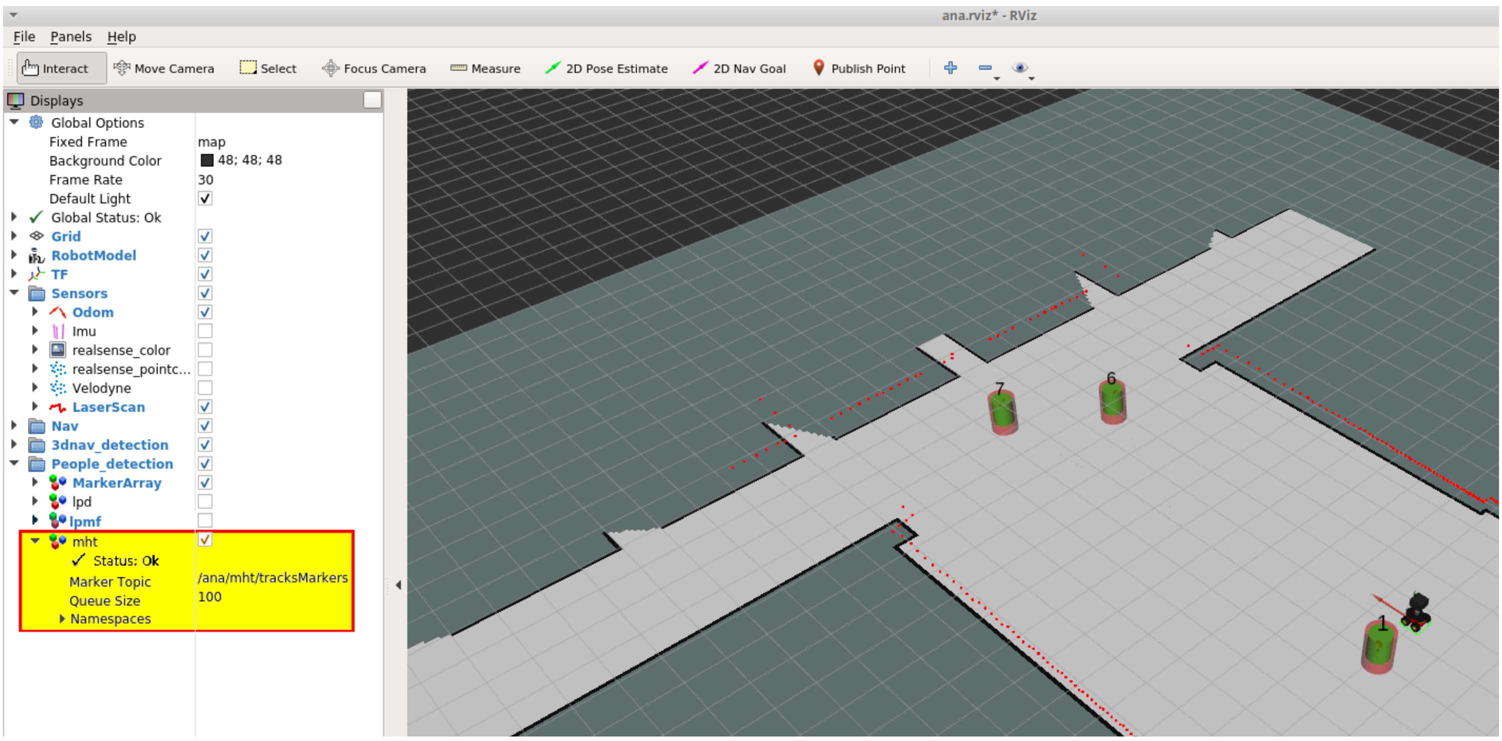
On the rviz you can disable the markers of the lase-people-detector by disabling the flag of these markers. The boolean of these markers is remarked in the next image (lpd). These markers are blue.



On the rviz you can disable the markers of the people-tracking by disabling the flag of these markers .The boolean of these markers is remarked in the next image (lpmf). These markers are yellow.



On the rviz you can disable the markers of the lase-people-filtering-using-the-map by disabling the flag of these markers, like the next image shows. The boolean of these markers is remarked in the next image (mht).



5. Parameters to customize the laser leg detector

5.1 Parameter description (rqt_reconfigure/lpd_2)

You can change/adapt several parameters of the detector from the Dynamic reconfigure parameters included in the cfg/LaserPeopleDetection.cfg of the node of ROS. Here, we include the description of all the parameters included in the Dynamic reconfigure. The actual set-up of the parameters that we include here are used with Dabo, which uses a Hokuyo 2D lidar.

- **RangeThreshold:** Distance in meters where to change use of classifier (near/far). Actual set-up to: 0.0 m.
- **detectionThreshold:** Classifying threshold which separates quality detections as positives or negatives. Actual set-up to: -0.07. If the robot are not able to detect some people, you may try to decrease this parameter to include the worst detections.
- **personRadius:** Radius to group pair of legs by. Actual set-up to: 0.7 m. Normally this parameter is between 0.5 m (for cluttered environments) and 0.7 m (for non cluttered ones).
- **JumpDistance:** Jump distance between points for segment generation. Actual set-up to: 0.1 m. The values of this parameter depend on the distance between two consecutive points of the laser beam detections. Normally, we use values inside the interval of 0.1 to 0.3 m
- **minPoints:** Minimum number of points for segment generation. Actual set-up to: 3.
- **selectPosesFrame:** Select frame where to publish detected poses (posesFrame param). Actual set-up to: False. In this case, we do not need to transform the detections to other frame.
- **PosesFrame:** Frame to transform poses to. Actual set-up to: /scan. In this case, we do not need to transform the detections to other frame.
- **SelectScanFrame:** Select frame where take scan messages from (scanFrame param). Actual set-up to: False . In this case, we do not need to transform the detections to other frame.
- **ScanFrame:** Only scans in scanFrame will be used. Actual set-up to: /scan. In this case, we do not need to transform the detections to other frame.
- **BoostFilePath:** path to classifier file (near), result of training. Actual set-up to: /usr/local/include/iri/laser_people_detection/boostData/boostFile.txt
- **boostFilePath2:** path to classifier file (near), result of training. Actual set-up to: /usr/local/include/iri/laser_people_detection/boostData/boostFile2.txt
- **markerWidth:** Dimension of detected people markers. Actual set-up to: 0.7 m.

- **markerHeight:** Dimension of detected people markers. Actual set-up to: 0.4 m.
- **markerR:** marker R color component. Actual set-up to: 0.0.
- **markerG:** marker G color component. Actual set-up to: 0.0.
- **markerB:** marker B color component. Actual set-up to: 0.5.
- **markerA:** marker ALPHA color component. Actual set-up to: 0.5.
- **filterPosesMode:** Filters poses by distance. Actual set-up to: False. Remember that you can filter the detections around the robot with this parameter, but initially this filtering is disabled.
- **FilterR:** Radius to filter poses out. Actual set-up to: 0.0 m. The filtering is disabled.
- **filterXmin:** min X coordinate to filter poses out. Actual set-up to: 0.0 m. The filtering is disabled.
- **FilterXmax:** max X coordinate to filter poses out. Actual set-up to: 0.0 m. The filtering is disabled.
- **FilterYmin:** min Y coordinate to filter poses out. Actual set-up to: 0.0 m. The filtering is disabled.
- **FilterYmax:** max Y coordinate to filter poses out. Actual set-up to: 0.0 m. The filtering is disabled.

6. Parameters to customize the map-filtering

6.1 Parameter description (rqt_reconfigure/lpaser_people_map_filter)

You can change/adapt several parameters of the map filter of detections from the Dynamic reconfigure parameters included in the cfg/LaserPeopleMapFilter.cfg of the node of ROS. Here, we include the description of all the parameters included in the Dynamic reconfigure. The actual set-up of the parameters that we include here are used with Dabo, which uses a Hokuyo 2D lidar.

- **Filter:** enable/disable map filtering. Enabled=True.
- **NeighborRadius:** Threshold distance in meters where to filter people detections. Normal set-up between : 0.25 or 0.3 m.
- **markerWidth:** Dimension of detected-filtered people markers. Actual set-up to: 0.7 m.
- **markerHeight:** Dimension of detected people markers. Actual set-up to: 1.5 m.
- **markerR:** marker R color component. Actual set-up to: 1.0.
- **markerG:** marker G color component. Actual set-up to: 1.0.
- **markerB:** marker B color component. Actual set-up to: 0.0.

- **markerA:** marker ALPHA color component. Actual set-up to: 0.9.

7. Parameters to customize the MHT tracker

7.1 Parameter that you may need to change to adapt the tracker to different environments and Laser-scans (rqt_reconfigure/mht_2)

You can change/adapt several parameters of the tracker from the Dynamic reconfigure parameters included in the cfg/PeopleTrackingMht.cfg of the node of ROS. Here, we include the description of all the parameters included in the Dynamic reconfigure. The actual set-up of the parameters that we include here are used with Dabo, which uses a Hokuyo 2D lidar.

- **max_dist_to_detect_people_conf:** It is the maximum radius around the robot that we allow to calculate the tracks. This radius can be all the range of the laser, 60 meters, or less. Actually, we set the parameter to 10 meters than is a considerable area around the person to track and predict the people position, and have enough time to avoid them or interact with them.). Actual set-up to: 60 m.

NOTE: This filtering radius do not work properly. It is better that you use the filtering radius of the detector, and filter around the robot position (10 meters or something similar).

- **frame_id:** frame to transform the input-detections and the output-tracks. for global tracker=map for local tracker=base_frame_of_your_robot. Actual set-up to: map.

- **local_tracker:** local_tracker=false for global tracker using the frame=map, and local_tracker=true for local tracker using the local frame of the robot. Actual set-up to: False.

- **association_threshold:** It is the maximum distance to associate detections with tracks. For our robot to track people, 1.5 m is a good vale, but you can change it. If you decrease it to 1 m for example you have less errors of changes of tracks due to crossing situations. If you increase it (for example to 4 m) you may deal with detections that change faster its position (more velocity change), and you can have more changes of tracks due to crossing situations. To track people, we normally oscillate between 1 m to 2 m for this parameter. Actual set-up to: 1.0 m.

- **confirmation_threshold:** It is the threshold of probability to confirm the tracks. We usually put it from 0.8 to 0.99. It depends on the frequency of obtain a good detection from the detector. Actual set-up to: 0.9.

- **deletion_threshold:** It is the threshold of probability to delete the tracks that do not have detection associated during certain period of time. We usually put it to 0.4, but it can oscillate between 0.5 to 0.4. Actual set-up to: 0.4.

- **laser_probability:** It is the probability of detection for the laser detections. For our system, it is set up to 0.9. This parameter can be change it, but we do not recommend it. It is better to change other parameters to customize the system. Actual set-up to: 0.9.

- **laser_false_alarm:** It is the probability of false alarm for the laser detections. For our system is set up to 0.1. This parameter can be change it, but we do not recommend it. It is better to change other parameters to customize the system. Actual set-up to: 0.1.

- **laser_new_track:** It is the probability of new track. laser_new_track=0.11. This parameter is highly recommend to do not change it. Actual set-up to: 0.11.

- **laser_no_detection:** To delete faster or slower the tracks, you have to change this parameter. It is the probability of no detection, with this parameter you control the slowest elimination of the tracks when the tracks do not have detection associated. For slowest deletion => laser_no_detection=0.999 (524 sec); for very fastest deletion laser_no_detection => 0.98 (30 sec) if you reduce more the value of this parameter you obtain less time for the deletion of the tracks. We use the laser_no_detection=0.97 to keep as much as possible the tracks to do not loss easily people that are interacting with the robot, but with more value, we keep more time the tracks. Actual set-up to: 0.97.

- **laser_no_confirmed:** It is the starting probability of laser no confirmed. laser_no_confirmed = 0.02. This parameter is highly recommended to do not change it. Actual set-up to: 0.02.

- **laser_no_confirmed_iteration:** To confirm faster or slower the tracks, you have to change this parameter. The value for the parameter that works for our system is: laser_no_confirmed_iteration=0.01, to filter as much false positive as we can. If you decrease the value of this parameter, the tracker expends more seconds to confirm the track; if you increase the value of this parameter , the tracker spends less seconds to confirm the track. laser_no_confirmed_iteration = 0.01 (the tracker spends 5.13 sec to confirm the track); for laser_no_confirmed_iteration = 0.0026 (the tracker spends 2 sec to confirm the track). Actual set-up to: 0.01.

- **augment_covariance_track:** It is the augment of the covariance if the track is not associated with any detection in the actual iteration. For us, the correct value is between [0.05 and 0.15] m. But you can increase it in situations where the tracks are moving faster to increase the association of the tracks with the detections. Actual set-up to: 0.5 m.

- **mht_velocity_margin:** It is the velocity margin to update well the kalman states of the tracks in two consecutive interactions. For our system used with people, it is set up to 1.0 m/s, to reduce errors due to changes in the detections if the detector only tracks one leg and the detection oscillate between the two legs, including an error in the velocity of the track. This margin allows only changes in the track velocity of 1.0 m/s between two iterations. Actual set-up to: 1.0 m/s.

- **covariance_no_track_conf:** It is the maximum value of the covariance allowed for the tracks until these will be removed. It is only to do not allow a very big area for the tracks when these tracks do not have detection associated. For our system this is set to covariance_no_track_conf=5.0 m. Also, you can reduce it, if you need it. If you see that the covariance of the tracks before removing it increases a lot, maybe it is good to reduce this parameter. Actual set-up to: 5.0 m.

The next parameters allow to enable the markers in the rviz visualization of ROS:

- **id_markers:** if true enable the visualization of the identifiers of the tracks over the tracks. Actual set-up to: True.

- **cov_markers:** if true enable enable to see the position of the tracks, visualized like green cylinders. Actual set-up to: True.

- **cov_markers_head:** if true enable enable to see a head over the tracks. Actual set-up to: False.

- **cov_markers_arm1:** if true enable enable to see a possible arm over the tracks. [These arms are oriented using the velocity, if the person is stop, maybe the position will be wrong, because the algorithm do not have the correct orientation of the person. Actual set-up to: False.

- **cov_markers_arm2:** if true enable enable to see the other arm of the person over the tracks. Actual set-up to: False.

- **vel_markers:** if true enable enable to see the velocity of the person, like a yellow arrow. Actual set-up to: True.

- **covdet_markers:** if true enable to see the position of the detections, like red cylinders biggest and with more transparency than the tracks. Actual set-up to: True.

- **covpred_markers:** if true enable to see the actual prediction of the tracks for the actual instant of time to be associated with the actual detection obtained, like a blue cylinder. Actual set-up to: False.

- **id_cluster_markers:** if true enable enable to see the identification numbers of the clusters. Each cluster are several tracks that are in crossing situations, where the tracks have to be split by proximity to the detections and using also the velocity of the track, to maintain it. Actual set-up to: False.

- **clust_markers:** if true enable to see the cluster markers. Actual set-up to: False.

- **group_markers:** if true enable enable to see the tracks of the groups. Each group are several tracks associated by proximity. This was introduced in the work with AGV (trucks of the port), where we detect several tracks, near and with the similar velocity, and we aggregate them, because some times the real objects were divided in several detections. Are represented by a green square marker. Actual set-up to: False.

- **track_path:** if true enable to see the before path of the tracks. Actual set-up to: False.

The last parameters were used to generate a database of only one track, obtained fusing several tracks obtained from different detectors obtained from different laser sensors localized and situated in the same environment.

We fused the information of the lasers of two of our robots using the combined rosbags, and we obtain the unique track of one person detected at the same time by the two different robots.

- **generate_database_companion:** If true enable to save the tracks in a txt file to do a database. Actual set-up to: False.

- **fuse_tibi_and_teo_tracks:** If true enable the fusion of two tracks detected by two sensors in one using the proximity between them. Actual set-up to: False.

- **debug_fuse_tibi_and_teo_tracks:** If true enable the debug messages of this track fusion. Actual set-up to: False.

7.2 Some recommendations about the use of the tracker and to know how to change well the parameters (rqt_reconfigure/mht_2)

If you have to use the tracker to other system and the results are not good, we recommend to read the publication [1] to understand the tracker behavior and change only the parameters: **association_threshold, confirmation_threshold, deletion_threshold, laser_no_detection, laser_no_confirmed_iteration, augment_covariance_track, mht_velocity_margin and covariance_no_track_conf.**

First, we recommend that in the detector, you decrease the maximum radius to detect people around the robot to reduce the tracking computational overload in huge and cluttered environments.

Second, respect the local_tracker mode. We recommend that, if you can, you chose the global_tracking mode, because the tracker works better in that mode. If not, you can use it in local_tracker mode, but you need to take into account different ways to filter moving objects, like we did in the paper of V. Vaquero, E. Repiso and A. Sanfeliu Sensors 2019.

Regarding the association_threshold. If you only have static people, with this parameter you chose the maximum range of the people detections to be associated with these people tracks. This parameter include movements due to the noise or static movements of the person or other errors in position of the detection system.

With the augment_covariance_track and mht_velocity_margin parameters, you control the association of the predictions of the tracks with the actual detections when the person is moving (in distance and in maximum change of velocity of these detections).

How to change the values of laser_no_confirmed_iteration and laser_no_detection is a little explained in this document and better explained in reference [1]. These are ones of the most important parameters to allow the good adaptation of the tracker behavior to the characteristics of your system. For more details about how to change these papers read the master thesis of Ely Repiso Polo, you can find it in her google scholar profile (https://scholar.google.es/citations?user=jO_K-WgAAAAJ&hl=es&oi=ao) or at her profile at the IRI web's page (https://www.iri.upc.edu/).

[1] Repiso Polo, E. (2015). Robust multi-hypothesis tracker fusing diverse sensor information (Master's thesis, Universitat Politècnica de Catalunya).

Also, if you use the software, please cite the following ([2]) publication.

[2] Vaquero, V., Repiso, E., & Sanfeliu, A. (2019). Robust and real-time detection and tracking of moving objects with minimum 2D LIDAR information to advance autonomous cargo handling in ports. Sensors, 19(1), 107.