UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Institut de Robòtica i Informàtica Industrial

People detection and tracking based on 2D range lidar

Detector

C++ library [1] to detect people using a Boosting based classifier. The person's leg data is coming from a 2D range lidar. An already trained classifier is provided in /data/classifierData/boostFile.txt, based on a dataset using a Hokuyo UTM-30LX sensor at ~0.4m height from the ground.

To use it with ROS, see its ROS wrapper package [2], allowing online running and showing an example using a recorded sequence.

[1] https://gitlab.iri.upc.edu/labrobotica/algorithms/**laser_people_detection**
[2] https://gitlab.iri.upc.edu/labrobotica/ros/perception/**iri_laser_people_detection**

To perform a new training, see laserPeopleTrain C++ library [3]. Also a ROS package labelling tool is provided [4], which allows visually labeling recorded sequence data.

[3] https://devel.iri.upc.edu/pub/labrobotica/algorithms/**laser_people_train**
[4] https://devel.iri.upc.edu/pub/labrobotica/ros/iri-ros-pkg_hydro/metapackages/iri_perception/**iri_laser_people_label**

Tracker

C++ library [5] to track people based on Kalman filter. It uses 2D points with (x,y) that can come from detections of lidar or other types of sensors. To use it with ROS, see the ROS wrapper package [6]. Also, you can only use the library in C++ to track points, but you have to make an executable program to do it.

The implementation follows a similar approach of the work presented in [Reid, 1979 [7]], and some of the contributions presented in [Luber et al, 2011 [8]]. Our implementation, was published in [Vaquero et al, 2015 [9]] and [Vaquero et al, 2019 [10]], and as a final-year degree project and master thesis in [Repiso, 2015 [11]].

The people tracker uses a Kalman filter to propagate the pedestrians' trajectories and it combines the different detections with the existing tracks Ids to calculate the most likely association hypothesis between them. The tracker uses hypothesis based on probabilities to confirm, hold, associate and delete the people tracks. Only time-consistent detections that are repeated multiple times become confirmed tracks. The algorithm can handle occlusions and crossings of detections. Furthermore, if we have an environment map, the system can filter the detections situated on a map, avoiding multiple false positive of person's leg detections. The leg detections are 2D points that corresponds to the central position of the person over the 2D plane. Then, the tracker can serve to track any kind of detections represented in 2D coordinates, also can be adapted to track detections in 3D coordinates.

If you use the software, please cite the following ([10]) publication.

[10] 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.

This software is usually used with the iri_laser_people_detection. All of the parameters of the tracker are correctly set up for this combination. If you use another type of detector or system, you have to set up the tracking parameters. An explanation of these parameters is included in the ReadMe.txt of the C++ library [5]. Note that in that case, you have also to set up the ROS launch file to use as input the topics of that detector.

Furthermore, we use this tracker and detector with a filter to remove the detections of the laser that are part of a predefined map of the environment. We do this to reduce the overload of the detector, with detections that are not really people. Then, It is recommended to do the same to obtain good results. To do it you can filter the detections that are very near to the map by proximity. Create this node is very easy or also you can ask for the node to the IRI technicians, because this filter node is not included with the iri_people_tracking_mht or iri_laser_people_detection.

Finally, you can find some additional explanations and how to install the packages in the ReadMe.txt included in the C++ library [5].

C++ Library:
[5] https://gitlab.iri.upc.edu/labrobotica/ros/perception/iri_people_tracking_mht/-/tree/master/**tracking_library_trunk**

ROS wrapper:
[6] https://gitlab.iri.upc.edu/labrobotica/ros/perception/**iri_people_tracking_mht**

References

[7] Reid, D. (1979). An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control, 24(6):843-854.
[8] Luber, M., Diego Tipaldi, G. & Arras, K. O. (2011). Place-dependent people tracking. The International Journal of Robotics Research, 30(3), 280-293.
[9] Vaquero, V., Repiso, E., Sanfeliu, A., Vissers, J., & Kwakkernaat, M. (2016). Low cost, robust and real time system for detecting and tracking moving objects to automate cargo handling in port terminals. In Robot 2015: Second Iberian Robotics Conference (pp. 491-502). Springer, Cham.
[10] 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.
[11] Repiso Polo, E. (2015). Robust multi-hypothesis tracker fusing diverse sensor information (Master's thesis, Universitat Politècnica de Catalunya). [https://upcommons.upc.edu/handle/2117/84172]