Towards a Robust People Tracking Framework for Service Robots in Crowded, Dynamic Environments

Timm Linder Fabian Girrbach Kai O. Arras

***Abstract*— People tracking is an important prerequisite for socially compliant service robots that operate in human environments. In this paper, we take this challenge to new extremes by attempting to robustly track people in a 360-degree field around the robot in very crowded environments like a busy airport terminal. As a first contribution, we present a novel multi-modal people tracking framework that is modular, flexible and fully integrated with ROS. We believe that our framework can be of great benefit to researchers as it covers the entire people tracking pipeline, including powerful visualization and evaluation tools. Secondly, we present a number of simple extensions that can make tracking in crowded scenarios more robust. Finally, we compare our tracking method against a complex multi-hypothesis tracking system. Our results on real and synthetic data suggest that it is not the choice of data association which has the largest impact, but the underlying models that, for instance, control track initiation, deletion, and handling of occlusions. By showing that the simpler data association may be sufficient, we provide reasoning why the available resources on a resource-constrained mobile robot might be better spent on other tasks such as higher-level perception and reasoning.**

I. Introduction

People tracking from a mobile platform in a first-person perspective has been studied in the robotics and computer vision communities for over a decade, and provides important functionality for assistance and service robots operating in human environments. It can lay the foundations to higher-level social processing such as socially aware motion planning [1], group detection and tracking [2], [3], social activity detection [4] or general human-robot interaction, and is crucial in person guidance tasks. The problem has basically been solved in simple scenarios with only a handful of persons walking in front of 2D laser [5] or RGB-D sensors [6], and good progress has been made in semi-crowded scenarios with 10–15 people simultaneously visible [7]–[10]. Few approaches are multi-modal in nature [11], [12].

However, larger numbers of persons need to be tracked when the entire 360-degree field of view around the robot shall be covered using an entire array of sensors, and not many systems have been evaluated in very complex and highly dynamic scenarios where over 30 persons can be present at the same time, such as in a very crowded airport terminal where our own service robot is going to be deployed (Fig. 1).

The authors are with the Social Robotics Lab, Dept. of Computer Science, University of Freiburg, Germany. http://srl.informatik.uni-freiburg.de, {linder,arras}@cs.uni-freiburg.de. This work has been partly supported by the European Commission under contract number FP7-ICT-600877 (SPENCER).

Fig. 1. Typical example of a crowded, highly dynamic situation in an airport terminal where we want to robustly and efficiently track persons with our service robot platform depicted on the bottom left.

In this paper, we present a multi-modal, highly modular and ROS-based people detection and tracking framework that, to our knowledge, is the most complete publicly available framework for this purpose, encompassing powerful visualization components, a group detection and tracking module, and implementations of different evaluation metrics for comparison with other approaches. Secondly, we discuss how a set of relatively simple extensions can make a person tracking system based upon very efficient nearest-neighbor (NN) data association more robust in challenging scenarios. We argue that the choice of data association method only plays a subordinate role, and what really matters are the models used to *e. g.* initiate and terminate tracks, handle occlusions, and predict human motion. We demonstrate this in a first set of experiments, where we compare our non-probabilistic nearest-neighbor method to a proven multi-hypothesis tracker that is significantly more complex in terms of implementation, parameter finding, and computational requirements. Our core tracking algorithm runs at less than 30% CPU load on a single core in complex scenarios, leaving enough computational resources for higher-level perception and social reasoning components.

Finally, we present a way of automatically tuning tracking parameters with regard to multi-target tracking metrics via an existing hyperparameter optimization library. Even in a simple NN-based system, there can be over 20 inter-dependent parameters that can affect tracking performance and require expert knowledge when tuned manually. These parameters are often part of noise models which significantly abstract

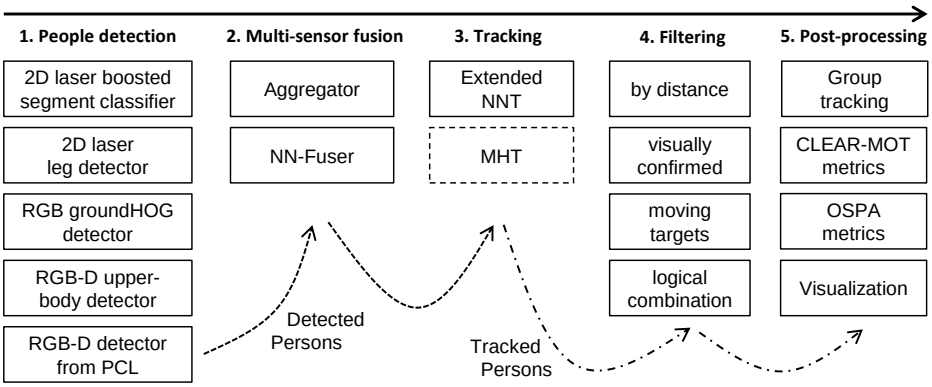

Fig. 2. Components and data flow between the 5 processing stages of our people tracking framework. All components are implemented as separate, reusable ROS modules, incorporating our own and third-party detection modules, tracking performance metrics, and powerful visualization components. All components, except for the one with dashed outline, are available as open-source.

from the underlying models of reality, and can at times be even counter-intuitive to use. Therefore, automated parameter tuning can significantly ease the burden of deploying our tracking framework in new scenarios.

II. Our framework

Fig. 2 gives an overview of the main components of our modular people tracking framework. All of these components are fully integrated into ROS and will be made publicly available on GitHub[1] at publication of this paper. In the following, we will briefly describe the most important components, starting from the left at the detection layer.

A. Detection

2D laser: For people detection in 2D laser range data, we have re-implemented a variant of the boosted laser segment classifier from [13] and integrated it with ROS. After a separate ROS node has segmented the laser scans using jump-distance clustering or a more accurate, but computationally more complex agglomerative hierarchical clustering (AHC), the detector computes a set of geometric 2D features on each segment which are then fed to the classifier. An additional high-recall blob detector which coarsely classifies the same kind of segments based upon their number of points and overall width has also been integrated.

We also integrated a publicly available leg detector[2] into our framework, which can provide better results if the sensor is mounted close to the ground such that both legs are visible as individual echoes in the laser scan. In our experiments with the sensor at 70–80 cm height, however, this detector (after parameter tuning[3], with the existing learned model) did not provide better results than our boosted segment classifier.

Monocular vision and RGB-D: For person detection in RGB-D, the existing depth template-based upper-body detector described in Jafari et al. [10], which runs in real-time at 20-30 Hz on the CPU, as well as their CUDA-based, monocular groundHOG detector have been integrated. We also extended the RGB-D person detector from [14], which applies a HOG classifier on candidate regions extracted from a depth-based height map, with GPU acceleration.

Fusing detections: For multi-sensor people tracking, our framework currently uses a flexible detection-to-detection fusion scheme configured fully via XML files. This allows to combine multiple sensor cues even when the particular tracking algorithm was not specifically designed to receive detection input from multiple sources. Using greedy nearest-neighbor association, we first fuse detections from sensors with overlapping fields of view (*e. g.* front laser, front RGB-D) and then aggregate the resulting sets of detections that do not overlap (*e. g.* front and rear detections).

All detectors integrated into our framework output detections which adhere to the same ROS message format. A *DetectedPerson* comprises a position vector $\mathbf{z}'$ and its uncertainty R' in a sensor-specific 3D coordinate frame, a scalar detection confidence, and some meta-data. R' can, for example, vary as a function of the person's distance to the sensor.

B. Tracking

In this section, we describe a new tracking system developed with robustness and computational efficiency in mind specifically for deployment on mobile service robots in crowded environments. Using a relatively cheap set of extensions from the target tracking community to systematically tackle shortcomings of current systems in such scenarios, we want to improve robustness without having to resort to multi-hypothesis tracking methods that are orders of magnitudes more complex in terms of implementation and computational requirements.

In our tracking system, detections arrive in their sensor-specific coordinate frame and are instantaneously transformed into a locally fixed frame (based upon robot odometry) that does not move with the robot. This ensures that the motion prediction of tracked persons is independent from the robot's ego-motion. In the resulting set of measurements $Z = \{\mathbf{z}_1, ..., \mathbf{z}_n\} \subset \mathbb{R}^2$, we drop the z coordinate as we currently only track in 2D world coordinates.

[1] https://github.com/spencer-project/spencer_people_tracking

[2] http://wiki.ros.org/leg_detector

[3] With original parameters, the detector had very low recall on our data.

Motion prediction: We predict the target motion by maintaining an Extended Kalman filter for each individual person. At our detection rate of around 30 Hz, human motion can in most scenarios be assumed to be locally linear, leading to the Nearly Constant Velocity model with state vector $\mathbf{x} = [x, \dot{x}, y, \dot{y}]^T$ and the transition matrix

$$F = \begin{bmatrix} 1 & t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

to predict a future state $\hat{\mathbf{x}} = F\mathbf{x}$, where t is the length of the tracking cycle. We use additive process noise Q to account for small changes in the velocity of the human motion. The process noise level q_{L} varies with the dynamics of the application scenario:

$$Q = \begin{bmatrix} \frac{1}{3}t^3 & \frac{1}{2}t^2 & 0 & 0 \\ \frac{1}{2}t^2 & t & 0 & 0 \\ 0 & 0 & \frac{1}{3}t^3 & \frac{1}{2}t^2 \\ 0 & 0 & \frac{1}{2}t^2 & t \end{bmatrix} q_{\mathrm{L}}. \tag{2}$$

Data association: Correctly associating new detections with existing tracks is crucial for good tracking performance. On the other hand, recent research [15] suggests that for people tracking from a mobile platform, with increasing environment complexity the difference in performance between various approaches such as Nearest-Neighbor (NN), Joint Probabilistic Data Association (JPDA) [5] and Probability Hypothesis Density (PHD) filters [15] diminishes.

Due to the significantly lower complexity that scales well with the number of tracked persons, we employ different variations of NN data association. The Global NN approach solves the complete linear assignment problem between detections and tracks using the Hungarian method [16] or the faster Jonkers-Volgenant [17] method, whereas Greedy NN searches for minima in the cost matrix in a greedy fashion. For all these approaches we use the Mahalanobis distance between track prediction and detection as assignment cost.

C. Track post-processing and visualization

Often, higher-level reasoning components are not interested in all tracks that are maintained internally by the people tracking system. Therefore, we provide a series of post-processing modules which filter the output such that it, for instance, only includes visually confirmed tracks, non-static persons, the n persons closest to the robot (useful for human-robot interaction), or a logical combination thereof. A standalone group detection and tracking module, based upon the coherent motion indicator features and the social network graph described in [3], has also been added.

The visualizations in this paper have been generated using a set of custom and highly configurable plugins for the ROS visualization tool *RViz* that are part of our framework.

III. Extensions for more robust tracking

A. Better motion prediction in dynamic scenarios

We provide a bank of first- and second order motion models, which cover different aspects of human motion. The variety of human motion has to be considered especially in crowded environments, where people are forced to change their motion according to the dynamics of the environment, which may lead to sudden stops or curvy trajectories. In addition to the Nearly Constant Velocity (CV) model (Eq. 2), the framework comprises the following motion models: Brownian Motion (Wiener Process), Nearly Constant Acceleration (CA) and Nearly Coordinated Turn (CT). Slight random variations of the constant term are modelled via additive white Gaussian noise. To capture the variety of human motion, the models may be combined inside an Interacting Multiple Models filter (IMM), which further helps to reduce the dependency on the process noise level q_L that strongly influences tracking performance but is a difficult to configure parameter due to the trade-off between tracking precision and robustness to sudden maneuvers.

B. Track initiation

Especially in sparse 2D laser range data, false positive detections can occur at high rates. To reduce the resulting number of 'ghost' tracks, we use a rule-based approach to confirm track creation. The *Track Initiation Logic* [18] was proven to perform well in a statistical and practical analysis for radar tracking [19]. For a track to be confirmed, it has to pass a gating step based upon the Euclidean distance between the measurement $\mathbf{z}$ and the current measurement prediction $H\hat{\mathbf{x}}$ of the track candidate, as well as a gating test that restricts the linear velocities to an interval $[v_{\min}, v_{\max}]$. If $v_{\min} > 0$, track initiation is restricted to moving targets, which prevents tracks being created from human-like objects such as trees or columns. As an extension to the original method, in each tracker cycle the velocity gating is performed recursively for each observation with regard to all observations already associated to the track candidate c_i, as shown in Alg. 1.

C. Track deletion logic

In the base version of our tracking system, we delete occluded tracks after a fixed number of tracking cycles n_{del} if no matching detection could be found. To prevent possible false alarm tracks from staying in the system for too long, we add a discrimination between 'young' and 'mature' tracks, where the latter is a track that over its lifetime has been matched at least n^{mat} times. In case a young track is occluded, it is deleted after being without detection for $n_{\mathrm{del}}^{\mathrm{yng}}$ cycles, whereas a mature track is deleted after a larger number of steps $n_{\mathrm{del}}^{\mathrm{mat}} > n_{\mathrm{del}}^{\mathrm{yng}}$ without detection.

IV. Experiments and Results

Since the focus of this paper is on tracking, and not detection, for the purpose of the following experiments we restrict ourselves to using 2D laser range data. The presented

Algorithm 1: Cascaded logic for track initiation

Data: unmatched detections Z^u, initiation candidates C
Params: min. match count $n_{\min}$, velocity thresholds
Result: new tracks T_{new} to be initiated
foreach *existing initiation candidate* $\mathbf{c}_i \in C$ **do**
  foreach $\mathbf{z}_j \in Z^u$ **do**
    Predict next measurement from state of $\mathbf{c}_i$
    Check distance between prediction and $\mathbf{z}_j$
    Check velocity between $\forall \mathbf{z} \in Z_{c_i}$ and $\mathbf{z}_j$
    if *checks passed* **then**
      $Z^u = Z^u \setminus \{\mathbf{z}_j\}$
      if $|Z_{c_i}| \geq n_{\min}$ **then**
        $C = C \setminus \{\mathbf{c}_i\}$
        Add $\mathbf{c}_i$ to T_{new}
      else
        Add $\mathbf{z}_j$ to $Z_{\mathbf{c}_i}$
        $n_{\text{miss},\mathbf{c}_i} = 0$
  if $\mathbf{c}_i$ *has no matching detection* $\mathbf{z} \in Z^u$ **then**
    $n_{\text{miss},\mathbf{c}_i} = n_{\text{miss},\mathbf{c}_i} + 1$
    if $n_{\text{miss},\mathbf{c}_i} > n_{\text{miss, max}}$ **then**
      Delete initiation candidate: $C = C \setminus \{\mathbf{c}_i\}$
foreach $\mathbf{z}_j \in Z^u$ **do**
  $C = C \cup$ new candidate $\mathbf{c}_k$ with $Z_{\mathbf{c}_k} := \{\mathbf{z}_j\}$

tracking method with its extensions is independent of the sensor modality used for detecting people.

A. Datasets

We evaluate the proposed system on two datasets of different complexity. Exemplary screenshots are shown in Fig. 3, with statistics on these two datasets in Table I. The first one is a popular 2D laser dataset recorded with a stationary SICK LMS-291 laser scanner at the Main Station in Freiburg, Germany. The second, new dataset is significantly more complex in terms of crowd dynamics and number of tracks within sensor range at a given time. This dataset has been synthetically generated using a combination of the pedestrian simulator *PedSim*, and the robot simulator *Gazebo*. Using our ROS wrapper of PedSim[4], we have modelled a scenario similar to the one our service robot will encounter in an airport terminal (Fig. 1), with large flows of people moving around corners towards specific goals, static groups and single persons spread all over the environment, and other people queuing up. Person interactions are modelled in PedSim using the social force model after Helbing [20], and pedestrian positions are then fed to Gazebo to continuously reposition 3D meshes in a scene in order to generate 2D laser scans via raycasting from a simulated mobile platform. While this approach obviously cannot correctly depict reality in all its detail, especially in terms of background complexity, it allows us to quickly study different scenarios, robot behaviors, and obtain exact groundtruth without need for manual annotations.

[4] https://github.com/srl-freiburg/pedsim_ros

	Frame Count		Track Count		
Dataset	Total	Annotated	Total	Avg	Max
Main Station	33,204	6,000	160	16.7	27
PedSim	4,965	4,965	90	78.6	90

TABLE I
DATASET STATISTICS

B. Evaluation metrics

A commonly used measure for evaluating multi-object tracking performance are the CLEAR-MOT metrics [21]. Besides counting false positives (FP), false negatives (FN) and ID switches (ID), they define an aggregate error measure called MOT Accuracy (MOTA) as

$$\text{MOTA} = 1 - \frac{\sum_k (\text{FP}_k + \text{FN}_k + \text{ID}_k)}{\sum_k \text{GT}_k},$$

where k is the tracking cycle index. The optimal MOTA score is 1.0, and MOTA can reach negative values if the tracker makes more errors than there are ground truth objects GT over the entire dataset duration.

As discussed extensively in [22], MOTA scores can vary between implementations and are dependent on meta-parameters such as the matching distance threshold θ_d and the way in which track hypotheses are assigned to groundtruth objects. Therefore, it is important to use the same metrics implementation when comparing different tracking approaches. While in our previous work [2], the tracking metrics were tightly integrated into the tracker, our framework provides a standalone Python implementation of CLEAR-MOT metrics as a separate ROS node that can be used to evaluate any kind of tracking algorithm compatible with our ROS message definitions. In our version, we compute groundtruth correspondences using a variant of the Hungarian method based upon Euclidean distances between object centroids in 2D world coordinates, with $\theta_d = 1\text{m}$.

Although MOTA can give a good impression of overall tracking performance, it is questionable if it alone can be a sufficient measure of tracking performance, as it weights all types of errors equally. Depending on the application scenario, some errors might have larger consequences (*e. g.* switching track IDs in a person guidance scenario). In our results, we therefore list these error types separately. Our framework also incorporates an implementation of the OSPA metrics [23], which we did not use in this work.

C. Experimental setup

All of our experiments were conducted on a high-end gaming laptop equipped with a quad-core Intel Core i7-4700 MQ processor and 8 GB of RAM under Ubuntu 14.04 with ROS Indigo. This is the actual computer platform used for this purpose on our service robot. For all experiments, we used the same boosted 2D laser segment classifier with agglomerative hierarchical clustering at a linkage threshold of 0.2m, pre-trained on annotated data from a SICK LMS-291 laser scanner at 75 cm height. We output detections with

Fig. 3. Example groundtruth tracks of the datasets used in our experiments, and originating laser endpoints (in grey). *Left:* Freiburg Main Station dataset, recorded with a stationary LMS 291 laser scanner. The camera image is only for visualization purposes. *Right:* Synthetic dataset generated by combining the pedestrian simulator *PedSim* with *Gazebo* to simulate 2D laser range data via raycasting (small pictures). Our simulation is modelled after a highly crowded, real airport scenario where people disembark from an airplane and enter the airport terminal. The simulated robot is driving through the $30\text{m} \times 30\text{m}$ scene for exploration and equipped with two laser scanners at 70 cm height.

a fixed position uncertainty of $R = \text{diag}(0.1, 0.1)$ up to a maximum range of 20m, after which laser echoes become very sparse and often consist of less than 3 points.

As an additional baseline method for comparison, we use a variant of the multi-hypothesis tracker (MHT) after Reid *et al.* [24] and Cox & Hingorani [25] with explicit occlusions labels [26]. We use n-scanback pruning with $n = 30$ and limit the maximum time $t_{\max}$ per tracking cycle in which an arbitrary number of hypotheses may be generated to 0.03s. The EKF parameters for the CV motion model are identical to the ones used for the NNT. The probabilities p_{det}, p_{occ}, p_{del} are 0.7, 0.27 and 0.03 as in [2]. The poisson rates for new tracks and false alarms were hand-tuned on our scenarios to $\lambda_{\text{new}} = 0.005$ and $\lambda_{\text{fal}} = 0.005$. We additionally enforce occluded tracks to be deleted after at most 10 cycles without detection, as they otherwise stay in the system for too long, leading to extremely bad MOTA scores in the very dynamic PedSim scenario.

D. Parameter tuning

Finding the correct parameter configuration is crucial to achieve good tracking results. Even in a simple NN-based system, there can be over 15 inter-dependent, performance-relevant parameters that often require expert knowledge for manual tuning. In particular, our experience is that i) the process noise level q_L, ii) the measurement covariances R, iii) track initiation- and iv) track deletion thresholds can have a high performance impact and are difficult to estimate.

We therefore integrated pySMAC[5], a Python wrapper for the hyperparameter optimization tool SMAC [27], with our extended NN tracker to identify well-performing parameter sets. SMAC allows to define categorical, integer and float parameter ranges and boundary conditions to be met. It uses a given performance metric, in our case MOTA, to fit a surrogate model in the form of a random forest. The model is used for prediction of promising configurations, which are then optimized using a combination of Bayesian optimization and local search.

V. Results

Table II shows tracking performance results of our extended NNT as well as the MHT baseline on the moderately crowded Freiburg Main Station and the very crowded PedSim datasets. We also show quantitative results of our tracking framework in different scenarios on our YouTube channel[6].

A. Data association

The results for the different data association methods, namely NN with multiple associations per detection, Global NN and Greedy NN, show that the sub-optimal solution of the assignment problem of the greedy method yields nearly the same results as the optimal solution found by applying the Jonkers-Volgenant [17] algorithm for the optimal solution. The NN variant that allows a detection to be assigned to multiple tracks yields higher results in MOTA, but also a higher number of ID switches. This can be explained by the fact that a track with a missing detection converges towards its nearest neighbor, which after a number of missed detections results in a track duplicate.

B. Track initiation logic

The track initiation logic by itself drastically reduces the FP rate by around half, but also leads to an increased miss rate for three different reasons. First, by requiring at least $n_{\min} = 6$ matches in our case for a track confirmation, each track's appearance is delayed by the same number of tracking cycles. Second, for targets outside of our velocity boundaries of $[v_{\min}, v_{\max}] := [0.2, 2]$, such as static persons, no tracks are initiated. Lastly, for targets that only infrequently trigger a detection, the number of consecutive allowed misses $n_{\text{miss, max}}$ might be too low.

C. Track deletion logic

Our track deletion logic has been tuned to delete young tracks after $n_{\text{del}}^{\text{yng}} = 20$ cycles and mature tracks after $n_{\text{del}}^{\text{yng}} = 50$ cycles. A track is being considered mature after at least $n^{\text{mat}} = 100$ matches. As can be seen from the results in Table II, this initially leads to a worse MOTA score compared to the baseline NN tracker due to the increase in false positives, as certain tracks are now allowed to survive for a longer time compared to the case without deletion logic where they are already deleted after $n_{\text{del}} = 5$ cycles.

However, once deletion and initiation logic are combined, the overall MOTA score increases because the two extensions

[5] https://github.com/automl/pysmac

[6] https://youtube.com/spencereuproject

Method	Main Station dataset MOTA	ID	FP%	Miss%	Hz	PedSim dataset MOTA	ID	FP%	Miss%	Hz
Global NN	71.9%	1280	8.4%	18.6%	6997	80.4%	4962	12.4%	5.7%	1323
Greedy NN, multi-association	70.7%	1439	9.8%	18.3%	6766	78.8%	5627	14.1%	5.5%	1403
Greedy NN	71.9%	1279	8.4%	18.6%	6976	80.5%	4968	12.4%	5.7%	1061
+Extended initiation logic	67.2%	781	2.5%	30.0%	7069	78.9%	3112	4.9%	15.3%	1260
+Deletion logic	66.7%	463	20.1%	12.7%	5732	71.1%	1842	25.0%	3.3%	747
+Base initiation logic	72.6%	274	6.6%	20.1%	6816	80.8%	1234	9.1%	9.6%	1025
+Extended initiation logic	73.3%	306	7.2%	19.3%	6472	82.2%	1315	9.4%	7.9%	935
+IMM (CV + CV)	73.3%	311	7.2%	19.3%	3433	82.2%	1272	9.4%	8.0%	606
MHT $t_{\max} = 0.03$s	72.2%	815	9.4%	17.4%	33	71.3%	3670	23.0%	4.8%	32

TABLE II

TRACKING PERFORMANCE COMPARISON

augment each other well. Combining the two, MOTA on the Main Station dataset rises from 71.9% to 73.3%, with an impressive reduction in ID switches from 1279 to 306. Similarly on the PedSim dataset, the number of ID switches is reduced by around 75%. Here, the extended version of the track initiation logic that recursively performs velocity gating against all previous detections that were already associated with the candidate, achieves 0.7-1.4% higher MOTA scores than the basic version which just performs velocity gating on the latest associated detection.

D. IMM

The IMM results in Table II were achieved using two CV models with different process noise levels $q_{\mathrm{L}_1} = 0.035$ and $q_{\mathrm{L}_2} = 0.267$. The parameters and the overall choice of motion models were found by automatic parameter optimization via PySMAC. While MOTA did not improve by adding the IMM, the number of ID switches went down slightly by 3.6% on the more challenging PedSim dataset. Additional qualitative experiments in our lab, where multiple persons interacted with our robot in a narrow environment, showed a reduction in track losses and subsequent ID switches. We believe that in such human-robot interaction scenarios, the IMM can lead to more obvious improvements, because the human subjects often try to 'play' with the robot and trick the tracking system into errors by performing erratic maneuvers. This happens less often in our recorded datasets, which do not include explicit human-robot interactions and therefore contain fewer sharp turns and persons stopping abruptly.

E. Comparison to other systems

Compared to the baseline nearest-neighbor tracker, the hypothesis-oriented MHT [26] – as expected – achieves better scores on the Main Station dataset[7]. On the highly challenging PedSim dataset, however, the MHT underperforms even after carefully tuning its parameters. We believe this to be due to the combinatorial explosion of possible data associations given the high track count[8]. This would necessitate a very high number of hypotheses, which is infeasible to maintain within the given cycle time limit of 30 ms. No improvement could be achieved by further raising the cycle limit to *e. g.* 100 ms.

Compared to the NNT with all extensions, the bare MHT performs 1% worse on the Main Station dataset, but 11% worse on the PedSim data. We believe the higher number of ID switches in MHT output, compared to the NNT, is due to frequent switching of the global best hypothesis. This is a well-known problem in multi-hypothesis tracking that is not straightforward to solve. For the sake of fairness, we want to note that we would expect tracking performance to increase by several percent if extensions such as the track initiation logic from Sec. V-B were also incorporated into the MHT.

F. Runtime performance

In the last column of Table II, we show the median of the extrapolated processing rates of the tracking algorithms based upon actually measured cycle times (without taking the detection stage into account). All methods have been hand-optimized for runtime performance. For an equal comparison, and with the application scenario of the service robot with limited on-board processing capabilities in mind, we restrict the tracker's CPU usage to a single thread.

It can be seen that the NNT, even with our extensions, is extremely efficient due to its simplicity, being able to theoretically process over 5000 tracking cycles per second in moderately crowded scenarios (Main Station), and still over 600 in very crowded scenarios (PedSim). The entire tracking framework, with 2 separate laser detectors for front and rear, runs in real-time at 35 Hz on our robot platform, with the tracker itself consuming less than 30% of a single CPU core even in crowded scenarios.

Looking at the cycle rates of the basic NNT on the PedSim dataset, which are in the order of around 1000 Hz, it easily becomes apparent why in such highly crowded scenarios with over 30 tracks in sensor range at a time, a hypothesis-oriented multi-hypothesis approach cannot succeed without massive parallelization. Since in the MHT, the data association step performed by the NNT needs to be repeated for every single hypothesis, we cannot expect more than $1000\,\mathrm{Hz} : 500\,\mathrm{hyp.} \approx 2\,\mathrm{Hz}$ on a single CPU core assuming

[7]These results are 8% worse than the baseline results reported in [2], because we track targets up to a maximum range of 20 meters (instead of 12m) and due to use of a different CLEAR-MOT implementation.

[8]In [26], for only four tracks up to 1000 hypotheses are generated.

we want to generate at least 500 hypotheses. Even on a processor capable of executing 8 threads in parallel, the expected frame rate would drop below 20 Hz without yet running any detection or higher-level perception components.

VI. Conclusion

In this paper, we have presented a modular, ROS-based framework for people tracking in crowded environments. As we have demonstrated by the integration of multiple person detectors, some of them from third-party sources, and two different tracking methods, our framework is easily extensible. We believe that our framework can be of benefit to researchers in service and assistance robotics, human-robot-interaction, and people tracking in particular, as it covers the entire people tracking pipeline, including powerful visualization and evaluation tools. Secondly, we have demonstrated that the choice of data association method matters less than expected. For practical applications on resource-constrained service robots, simple data association methods combined with effective extensions like a track initiation logic can be a better choice than highly complex multi-hypothesis approaches.

In future work, we want to extend our evaluation to multi-modal sensor data and compare quantitatively against further publicly available state-of-the art tracking methods (e. g. [10]) on challenging data captured in a crowded airport environment (Fig. 1).

To resolve data association ambiguity during extended occlusions, we plan to integrate appearance-based cues whenever persons enter the field of view of an RGB-D sensor on our robot. In these cases, we could also inhibit the track initiation logic to allow detection of standing persons, if the RGB-D detector is sufficiently confident. Finally, we want to integrate other promising methods from the literature that have, to our knowledge, never been combined in a single system, including feedback of group information into person-level tracking [2], motion prediction informed by a social force model [8], and occlusion geodesics [28].

References

[1] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013.

[2] M. Luber and K. O. Arras, "Multi-hypothesis social grouping and tracking for mobile robots," in *Proceedings of Robotics: Science and Systems*, Berlin, Germany, 2013.

[3] T. Linder and K. O. Arras, "Multi-model hypothesis tracking of groups of people in RGB-D data," in *IEEE Int. Conf. on Information Fusion (FUSION'14)*, Salamanca, Spain, 2014.

[4] B. Okal and K. O. Arras, "Towards group-level social activity recognition for mobile robots," in *IROS 2014 Workshop on Assistance and Service Robotics in a Human Environment*, Chicago, USA, 2014.

[5] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *The International Journal of Robotics Research*, vol. 22, no. 2, pp. 99–116, 2003.

[6] F. Basso, M. Munaro, S. Michieletto, E. Pagello, and E. Menegatti, "Fast and robust multi-people tracking from RGB-D data for a mobile robot," in *Intelligent Autonomous Systems 12*, ser. Advances in Intelligent Systems and Computing, S. Lee, H. Cho, K.-J. Yoon, and J. Lee, Eds. Springer Berlin Heidelberg, 2013, vol. 193, pp. 265–276.

[7] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[8] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'10)*, Anchorage, USA, 2010.

[9] M. Luber, L. Spinello, and K. O. Arras, "People tracking in RGB-D data with online-boosted target models," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, San Francisco, USA, 2011.

[10] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

[11] C. Martin, E. Schaffernicht, A. Scheidig, and H.-M. Gross, "Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking," *Robotics and Autonomous Systems*, vol. 54, no. 9, pp. 721 – 728, 2006, selected papers from the 2nd European Conference on Mobile Robots (ECMR 2005) 2nd European Conference on Mobile Robots.

[12] N. Bellotto and H. Hu, "Multisensor-based human detection and tracking for mobile service robots," *IEEE Trans. on Systems, Man, and Cybernetics – Part B*, vol. 39, no. 1, pp. 167–181, 2009.

[13] K. O. Arras, O. Martínez Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Proc. of the Int. Conf. on Robotics & Automation*, 2007.

[14] M. Munaro, F. Basso, and E. Menegatti, "Tracking people within groups with RGB-D data," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct 2012.

[15] J. Correa, J. Liu, and G.-Z. Yang, "Real time people tracking in crowded environments with range measurements," in *Social Robotics*, ser. Lecture Notes in Computer Science, G. Herrmann, M. Pearson, A. Lenz, P. Bremner, A. Spiers, and U. Leonards, Eds. Springer International Publishing, 2013, vol. 8239, pp. 471–480.

[16] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, pp. 83–97, 1955.

[17] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, 1987.

[18] Y. Bar-Shalom, *Tracking and Data Association*. San Diego, CA, USA: Academic Press Professional, Inc., 1987.

[19] Z. Hu, H. Leung, and M. Blanchette, "Statistical performance analysis of track initiation techniques," *Signal Processing, IEEE Transactions on*, vol. 45, no. 2, pp. 445–456, Feb 1997.

[20] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, May 1995.

[21] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the CLEAR MOT metrics," *Journal of Image Video Processing*, vol. 2008, 2008.

[22] A. Milan, K. Schindler, and S. Roth, "Challenges of ground truth evaluation of multi-target tracking," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, June 2013, pp. 735–742.

[23] B. Ristic, B.-N. Vo, D. Clark, and B.-T. Vo, "A metric for performance evaluation of multi-target tracking algorithms," *Signal Processing, IEEE Transactions on*, vol. 59, no. 7, pp. 3452–3457, July 2011.

[24] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. on Automatic Control*, vol. 24, no. 6, 1979.

[25] I. Cox and S. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 18, no. 2, 1996.

[26] K. O. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, USA, 2008.

[27] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Learning and Intelligent Optimization*. Springer, 2011, pp. 507–523.

[28] H. Possegger, T. Mauthner, P. Roth, and H. Bischof, "Occlusion geodesics for online multi-object tracking," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1306–1313.