Range-Based People Detection and Tracking for Socially Enabled Service Robots

Kai O. Arras, Boris Lau, Slawomir Grzonka, Matthias Luber,
Oscar Martinez Mozos, Daniel Meyer-Delius, and Wolfram Burgard

Abstract. With a growing number of robots deployed in populated environments, the ability to detect and track humans, recognize their activities, attributes and social relations are key components for future service robots. In this article we will consider fundamentals towards these goals and present several results using 2D range data. We first propose a learning method to detect people in sensory data based on a set of boosted features. The method largely outperforms the state of the art that typically relies on hand-tuned classifiers. Then, we present a person tracking approach based on the detection and fusion of leg tracks. To deal with the frequent occlusion and self-occlusion of legs, we extend a Multi-Hypothesis Tracking (MHT) approach by the ability to explicitly reason about and deal with adaptive occlusion probabilities. Finally, we address the problem of tracking groups of people, a first step towards the recognition of social relations. We further extend the MHT approach by a multiple model hypothesis stage able to reflect split/merge events in group formation processes. The proposed extension is mathematically elegant, runs in real-time and further allows to accurately estimate the number of people in each group. The article concludes with prospects and suggestions for future research.

1 Introduction

In many applications, service robots share their space with people. This makes people detection and tracking, navigation in populated environments, and human-robot interaction key problems in the context of this book.

Kai O. Arras · Boris Lau · Slawomir Grzonka · Matthias Luber · Oscar Martinez Mozos · Daniel Meyer-Delius · Wolfram Burgard
Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Germany
e-mail: {arras,lau,luber,grzonka}@informatik.uni-freiburg.de,
{meyerdel,burgard}@informatik.uni-freiburg.de,
e-mail: omozos@irvs.ait.kyushu-u.ac.jp

E. Prassler et al. (Eds.): Towards Service Robots for Everyday Environ., STAR 76, pp. 235–280.
springerlink.com

The problem statement of people detection is to identify which sensor readings originate from humans and which measurements belong to other dynamic objects, background, or clutter. The problem statement of tracking is – based on detection hypotheses – to estimate position and motion state of people, robustly handle occlusions, and to correctly associate measurements to tracks accounting for false alarms and newly arriving targets. Traditionally, target tracking has been studied for air- or water-borne targets using radar or sonar. In this article, we address the problem of people tracking in range data as most service robots are equipped with laser range finders. Range as a sensor modality has a number of important advantages that makes it an appropriate choice for service robots in real-world applications. Unlike cameras, range finders provide accurate and robust depth information with high resolution. They are reliable under a wide range of lighting conditions and have a large field of view. Finally, and especially in situations with little light, range finders are robust against vibrations from a moving vehicle.

In the first section of the article we consider the problem of people detection in 2D range data using a boosted classifier for segments of adjacent laser points [1]. The positively predicted segments are considered to be human legs, and each segment is treated as a single leg measurement by the tracking system. The tracker, which is presented in the second section, uses assignment solving to either associate these legs with existing tracks, or to label them as new tracks or false alarms. Tracks that are not matched by observations are labeled as occluded or deleted. The probability of a global assignment is computed as the product of the likelihoods of the individual assignments of all tracks and measurements [2].

We employ the Multiple-Hypothesis Tracking framework by Reid [3] to increase tracking robustness: by not only assuming the most likely assignment as the correct one but explicitly maintaining the n-best solutions, data association decisions are integrated over time. The leg tracks are then assembled to person tracks by using a simplistic person model. This allows the tracker to account for the increased self-occlusion probability of the associated leg tracks. This way, robust tracking is achieved.

If many people are present in an environment, tracking individuals can become a very difficult task. Both detection and data association of people is becoming increasingly hard in densely crowded environments. Further, in many applications of service robots, people are encountered in groups and not as individuals. This raises a different perspective and motivates tracking of groups of people. Tracking groups, subsuming the state of several people into a single group state, greatly simplifies data association and allows to recognize interrelationships between people. The latter point is crucial for effective and socially compliant robot behavior, for instance, when approaching and addressing groups or when moving among groups. Social relations of people cannot be perceived directly by the sensors of a robot. However, according to the Proxemics theory by Hall [4], their relations are correlated with spatial distance during interaction, which is observable for the robot. The third section in this article presents an approach to track the joint state of each group of people and follows group formation processes over time [5]. This way information about their social relation can be recovered robustly even during difficult situations, e.g., when people from different groups come very close.

2 Leg Detection in 2D Range Data

In this section, we consider the problem of people detection from data acquired with laser range finders. The application of such sensors for this task has been popular in the past for the above mentioned reasons—accuracy and robustness over a wide range of ambient conditions. However, laser range data contain little information about people, especially because they typically consist of two-dimensional range information. Figure 1 shows an example scan from a cluttered office environment. While this scan was recorded, several people walked through the office. The scan suggests that in cluttered environments, people detection in 2D is difficult even for humans. However, at a closer look, range measurements that correspond to humans have certain geometrical properties such as size, circularity, convexity or compactness (see Figure 2). The key idea of our approach is to determine a set of meaningful scalar features that quantify these properties and to use supervised learning to create a people detector with the most informative features. In particular, it employs AdaBoost as a method for selecting the best features and thresholds, while at the same time creating a classifier using the selected features.

In the past, many researchers focused on the problem of tracking people in range scans. One of the most popular approach in this context is to extract legs by detecting moving blobs that appear as local minima in the range image [6, 7, 8, 9]. To this end, two types of features have been quite popular: motion and geometry features. Motion in range data is typically identified by subtracting two subsequent scans. If the robot is moving itself, the scans have first to be aligned, e.g., using scan matching. The drawback of motion features is that only *moving* people can be found. Topp and Christensen [10] extend the method of Schulz *et al.* [9] by the ability to track also people standing still, which, for instance, is useful for interaction. They report on good results in typical scenarios but also on problems in cluttered environments. They also conclude that either improved motion models or more advanced pattern detection of people are necessary.

Cui *et al.* [11] pursue a multi-sensor approach to people tracking using multiple laser scanners at foot height and a monocular camera. After registration of the laser data they extract moving blobs of 15 cm diameter as feet candidates. Two feet candidates at a distance of less than 50 cm are treated as a step candidate.

Geometric features have also been used by Xavier *et al.* [12]. With a jump distance condition they split the range image into clusters and apply a set of geometric rules to each cluster to distinguish between lines, circles and legs. A leg is defined as a circle with an additional diameter condition.

In all approaches mentioned above, neither the selection of features nor their thresholds are learned or determined other than by manual design and hand-tuning. This motivates the application of a learning technique.

Hähnel *et al.* [13] have considered the problem of identifying beams in range scans that are reflected by dynamic objects. They consider the individual beams independently and apply EM to determine whether or not a beam has been reflected by a dynamic object such as a person. Our method, in contrast, considers groups of beams and classifies the entire groups according to their properties.

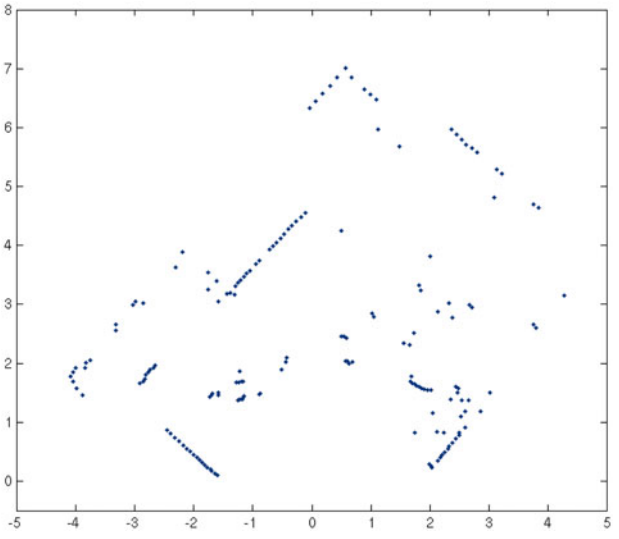

Fig. 1 Where are the people? Example scan from a typical office.

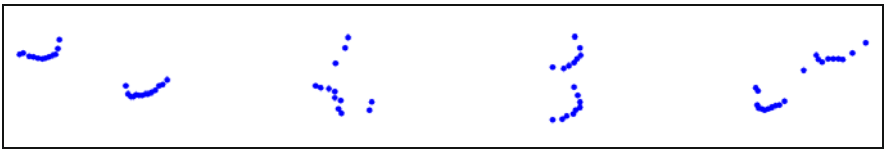

Fig. 2 Typical range readings from legs of people. As can be seen, the appearance can change drastically, also because the legs cannot always be separated. Accordingly, the proper classification of such pattern is difficult.

AdaBoost has been successfully used as a Boosting algorithm in different applications for object recognition. Viola and Jones [14] boost simple features based on grey level differences to create a fast face classifier using images. Treptow *et al.* [15] use the AdaBoost algorithm to track a ball without color information in the context of RoboCup. Further, Martínez Mozos *et al.* [16] apply AdaBoost to create a classifier able to recognize places in 2D maps. They use a set of geometrical features extracted from range data as input for the boosting algorithm. Also Rottmann *et al.* [17] use geometrical features together with vision features as input for AdaBoost. The vision features are based on the number of certain type of objects detected in an image.

Our motivation is the belief that the definition of appropriate features for the detection of people in range data has been underestimated as a problem so far. In the context of people tracking, the focus has mostly been on the tracking algorithms rather than on the feature detection problem. We believe that a more reliable feature detection will ultimately improve tracking performance.

2.1 Boosting

Boosting is a general method for creating an accurate strong classifier by combining a set of weak classifiers. The requirement to each weak classifier is that its accuracy is better than a random guessing. In this work we use the AdaBoost algorithm introduced by Freund and Schapire [18]. The input to the algorithm is a set of labeled training data $(e_n, l_n), n = 1, \ldots, N$, where each e_n is an example and $l_n \in \{+1, -1\}$ indicates whether e_n is positive or negative respectively. In a series of rounds $t = 1, \ldots, T$, the algorithm repeatedly selects a weak classifier $h_t(e)$ using a weight distribution D_t over the training examples. The selected weak classifier is expected to have a small classification error in the weighted training examples. The idea of the algorithm is to modify the distribution D_t at each round: it increases the weights of the examples which were incorrectly classified by the previous weak classifier. The final strong classifier H is a weighted majority vote of the T best weak classifiers. Large weights are assigned to good weak classifiers whereas poor ones receive small weights.

We use the approach presented by Viola and Jones [14] in which the weak classifiers depend on single-valued features f_j and have the form

$$h_j(e) = \begin{cases} +1 \text{ if } p_j f_j(e) < p_j \theta_j \\ -1 \text{ otherwise,} \end{cases} \tag{1}$$

where θ_j is a threshold and p_j is either $+1$ or -1 and thus represents the direction of the inequality. In each round t of the algorithm, the values for θ_j and p_j are learned, so that the misclassification in the weighted training examples is minimized [16]. The final AdaBoost algorithm modified for the concrete task of leg-detection in range data is shown in Table 1.

Table 1 The generalized AdaBoost algorithm

- Input: Set of examples $(e_1, l_1), \ldots, (e_N, l_N)$, where $l_n = +1$ for positive examples and $l_n = -1$ for negative examples.
- Initialize weights $D_1(n) = \frac{1}{2a}$ for $l_n = +1$ and $D_1(n) = \frac{1}{2b}$ for $l_n = -1$, where a and b are the number of positive and negative examples respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights: $D_t(n) = \frac{D_t(n)}{\sum_{i=1}^{N} D_t(i)}$.
  2. For each feature f_j train a weak classifier h_j using D_t.
  3. For each h_j calculate: $r_j = \sum_{n=1}^{N} D_t(n) l_n h_j(e_n)$, where $h_j(e_n) \in \{+1, -1\}$.
  4. Choose h_j that maximizes $|r_j|$ and set $(h_t, r_t) = (h_j, r_j)$.
  5. Update the weights: $D_{t+1}(n) = D_t(n) \exp(-\alpha_t l_n h_t(e_n))$, where $\alpha_t = \frac{1}{2} \log(\frac{1+r_t}{1-r_t})$.
- The final strong classifier is given by: $H(e) = \text{sign}(F(e))$, where $F(e) = \sum_{t=1}^{T} \alpha_t h_t(e)$.

2.2 Feature Definitions

In this section we describe the segmentation method and the features used in our system. We assume that the robot is equipped with a range sensor that delivers observations $Z = \{b_1, ..., b_L\}$ that consist of a set of beams. Each beam b_j corresponds to a tuple (ϕ_j, ρ_j), where ϕ_j is the angle of the beam relative to the robot and ρ_j is the length of the beam.

The beams in the scan Z are split into subsets of beams based on a segmentation algorithm. In our current system, we use a jump distance condition to compute the segmentation: If two adjacent beams are farther away than a threshold distance, a new subset is initialized. Although one could easily imagine more complex or adaptive thresholds (see the work by Premebida and Nunes [19] for an overview), we found in our experiments that the jump distance condition yields segmentations that can readily be processed by the subsequent learning step.

The output of the partitioning procedure is an ordered sequence $\mathcal{P} = \{S_1, S_2, ..., S_M\}$ of segments such that $\bigcup S_i = Z$. The elements of each segment $S = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ are represented by Cartesian coordinates $\mathbf{x} = (x, y)$, where $x = \rho\cos(\phi)$ and $y = \rho\sin(\phi)$, and (ϕ, ρ) are the polar coordinates of the corresponding beam.

The training examples for the AdaBoost algorithm are given by a set of segments together with their labels

$$E = \{(S_i, l_i) \mid l_i \in \{+1, -1\}\},$$

where $l_i = +1$ indicates that the segment S_i is a positive example and $l_i = -1$ indicates that the segment S_i is a negative example. Note that the standard AdaBoost algorithm is a binary classifier only. In situations,F in which there are different objects to be classified, one could learn decision lists as successfully applied by Martínez Mozos *et al.* [16] in the context of place labeling with mobile robots.

We define a feature f as a function $f : \mathcal{S} \to \Re$ that takes a segment S as an argument and returns a real value. Here, $\mathcal{S}$ is the set of all possible segments. For each segment S_i we determine the following fourteen features:

1. *Number of points*: $n = |S_i|$.
2. *Standard deviation*: This feature is given by

$$\sigma = \sqrt{\frac{1}{n-1}\sum_j ||\mathbf{x}_j - \bar{\mathbf{x}}||^2},$$

   where $\bar{\mathbf{x}}$ denotes the center of gravity of a segment S_i.
3. *Mean average deviation from median*: This feature is designed to measure the segment compactness more robustly than the standard deviation. The median of a distribution $f(x)$ is the value where the cumulative distribution function $F(x) = 1/2$. Given an *ordered* set of K scalar random samples x_i the median $\tilde{x}$ is defined as

$$\tilde{x} = \begin{cases} x_{(K+1)/2} & \text{if } K \text{ is odd} \\ \frac{1}{2}(x_{K/2} + x_{K/2+1}) & \text{if } K \text{ is even} \end{cases}$$

Opposed to the mean, the median is less sensitive to outliers. In our multi-dimensional case, we calculate $\tilde{\mathbf{x}}$ using the vector-of-medians approach [20], i.e. $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})$. The average deviation from the median is then

$$\varsigma = \frac{1}{n} \sum_j ||\mathbf{x}_j - \tilde{\mathbf{x}}||$$

4. *Jump distance from preceeding segment*: This feature corresponds to the Euclidian distance between the first point of S_i and the last point of S_{i-1}.
5. *Jump distance to succeeding segment*: The Euclidian distance between the last point of S_i and the first point of S_{i+1}.
6. *Width*: This feature measures the Euclidian distance between the first and last point of a segment.
7. *Linearity*: This feature measures the straightness of the segment and corresponds to the residual sum of squares to a line fitted into the segment in the least squares sense. Given the segment points in polar coordinates $\mathbf{x}_i = (\phi, \rho)$, fitting a line in the Hessian (α, r)-representation that minimizes perpendicular errors from the points onto the line has a closed form solution. We use the (unweighted) expressions from [21]. Once the line parameters (α, r) are found, the residual sum of squares is calculated as

$$s_l = \sum_j (x_j cos(\alpha) + y_j sin(\alpha) - r)^2,$$

where $x_j = \rho_j cos(\phi_j)$ and $y_j = \rho_j sin(\phi_j)$.

8. *Circularity*: This feature measures the circularity of a segment. Like for the previous feature, we sum up the squared residuals to a fitted circle. Given a set of points in Cartesian coordinates, an elegant and fast way to find the best circle in the least squares sense is to parameterize the problem by the vector of unknowns as $x = (x_c \quad y_c \quad x_c^2 + y_c^2 - r_c^2)^T$ where x_c, y_c and r_c denote the circle center and radius. With this, the overdetermined equation system $A \cdot x = b$ can be established,

$$A = \begin{pmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ -2x_n & -2y_n & 1 \end{pmatrix} \quad b = \begin{pmatrix} -x_1^2 - y_1^2 \\ -x_2^2 - y_2^2 \\ \vdots \\ -x_n^2 - y_n^2 \end{pmatrix}$$

and solved using the pseudo-inverse

$$x = (A^T A)^{-1} A^T \cdot b.$$

The residual sum of squares is then

$$s_c = \sum_{i=1}^{n} (r_c - \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2})^2.$$

This parameterization of the least squares problem has better geometric properties than the approach used by Song *et al.* [22]. When geometry plays a role in fitting (opposed, e.g., to regression in statistics), care has to be taken what errors are minimized. Otherwise algebraically correct but geometrical useless least squares fits can be the result.

9. *Radius*: This feature is the radius r_c of the circle fitted to the segment. It corresponds to an alternative measure of the size of a segment S_i.
10. *Boundary length*: This feature measures the length

$$l = \sum_j d_{j,j-1}$$

of the poly-line corresponding to the segment, where $d_{j,j-1} = \|\mathbf{x}_j - \mathbf{x}_{j-1}\|$ is the distance between two adjacent points in the segment.
11. *Boundary regularity*: Here we calculate the standard deviation of the distances $d_{j,j-1}$ of adjacent points in a segment.
12. *Mean curvature*: The average curvature $\bar{k} = \sum \hat{k}_j$ over the segment S_i is calculated using the following curvature approximation. Given a succession of three points $\mathbf{x_A}$, $\mathbf{x_B}$, and $\mathbf{x_C}$, let A denote the area of the triangle $\mathbf{x_A x_B x_C}$ and d_A, d_B, d_C the three distances between the points. Then, an approximation of the discrete curvature of the boundary at $\mathbf{x_B}$ is given by

$$\hat{k} = \frac{4A}{d_A d_B d_C}.$$

This is an alternative measurement of r_c as curvature and radius are inverse proportional.
13. *Mean angular difference*: This feature traverses the boundary and calculates the average of the angles β_j between the vectors $\overline{\mathbf{x}_{j-1}\mathbf{x}_j}$ and $\overline{\mathbf{x}_j\mathbf{x}_{j+1}}$ where

$$\beta_j = \angle(\overline{\mathbf{x}_{j-1}\mathbf{x}_j}, \overline{\mathbf{x}_j\mathbf{x}_{j+1}}).$$

Care has to be taken that angle differences are properly unwrapped. This features is a measure of the convexity/concavity of segment S_i.
14. *Mean speed*: Given two scans with their associated timestamps T_k, T_{k+1}, this feature determines the speed v_j for each segment point along its beam,

$$v_j = \frac{\rho_j^{k+1} - \rho_j^k}{T_{k+1} - T_k},$$

and averages over all beams in the segment. ρ_j^k and ρ_j^{k+1} are the range values of beam j at times k and $k+1$.

This collection of features constitutes a profile of each segment (see Figure 3). Since certain features are not defined for less than three points (e.g., circularity, radius) only segments with $n > 2$ points are taken into account.

2.3 Experimental Evaluation

The approach presented above has been implemented using a 180 degree SICK laser range finder. The goal of the experiments is to demonstrate that our simple features can be boosted to a robust classifier for the detection of people. Throughout the experiments, the sensor was kept stationary and mounted 30 cm above the floor. The corresponding scans where segmented and the features described in Section 2.2 were calculated for each segment. The complete set of labeled segments was then divided randomly into a training and a test set, each containing approximately 50% of the segments. The training sets were employed for learning a strong classifier using AdaBoost, whereas the test set was used for the evaluations. The segments in the test sets were labeled manually as person or non-person. With the help of videos recorded during the experiment, the ground truth could be properly identified.

We first demonstrate how our classifier can be learned to detect people in two different environments, namely a corridor and an office. Additionally we analyze whether a common classifier can be used in both environments. Further we show

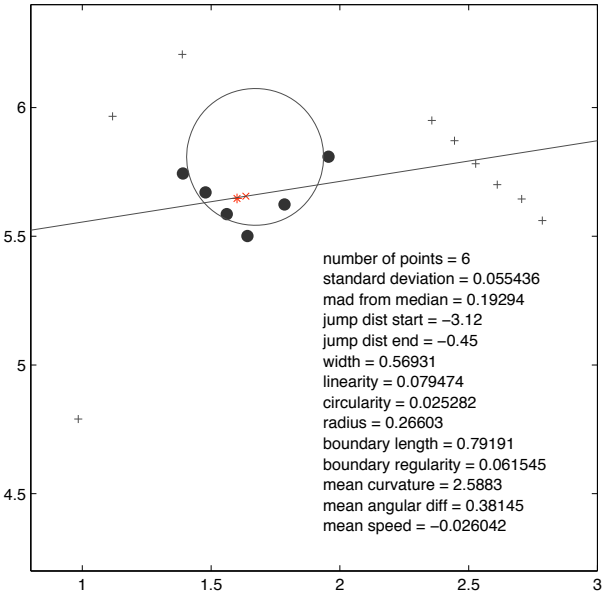

Fig. 3 Laser segment with its feature profile. The highlighted points correspond to the segment and the crosses depicts other readings in the scan. The circle and line are fitted to the segment for the *linearity* and *circularity* features.

Fig. 4 The corridor (left) and office (right) environments in which the experiments were carried out.

how a classifier can be used to classify people in environments for which no training data were available. We also compare our results with the ones obtained using a heuristic approach which uses features frequently found in the literature about laser-based people tracking. In all these experiments we applied features #1 to #13 from Section 2.2. In a final experiment we repeat the training and classification steps using also the motion feature #14.

One important parameter of the AdaBoost algorithm is the number of weak classifiers T used to form the final strong classifier. We performed several experiments with different values for T and we found that $T = 10$ weak classifiers provide the best trade-off between the error rate of the classifier and the computational cost of the algorithm.

2.3.1 Corridor and Office Environments

In the first experiment we recorded a total of 540 scans in a corridor while a person was both moving and standing still (Figure 4 left). Each scan was divided into segments and for each segment features #1 to #13 were calculated. The total number of segments extracted was 5734. After dividing the segments into a training and a test set, we trained our AdaBoost classifier. The results from the test set are shown in Table 2. Only 1 from 240 segments (0.42%) corresponding to people was misclassified (false negatives), whereas 27 from 2589 segments (1.03%) not corresponding to a person were classified as people (false positives). These results show that our algorithm can detect people with high accuracy.

In a second experiment we placed the laser in an office that contained tables, chairs, boxes, round shaped trash bins, and other furniture, creating a cluttered environment. Figure 4 (right) shows a picture and Figure 1 depicts a scan taken in this office. Two people were in the room during the experiment. Like in the previous experiment, the people were moving and occasionally standing still. A total of 791 scans were recorded from which we extracted 13838 segments. The segments were divided into a training and a test set and a strong classifier was learned. Although

the office was cluttered with objects and furniture that strongly resemble features of legs, we still obtained an overall classification rate of 97.25%. The confusion matrix is shown in Table 3.

In a third experiment we created a common set of segments containing all the segments from both the corridor and the office environment. Again, the set was divided into a training and a test set. Table 4 shows the confusion matrix. Although the error rates slightly increase with respect to Tables 2 and 3, they still remain under 4%, which in our opinion is a fairly good level. This result demonstrates that a common classifier can be learned using both environments while still obtaining good classification rates.

2.3.2 Transferring the Classifiers to New Environments

The following experiment was designed to analyze whether a classifier learned in a particular environment can be used to successfully detect people in a new environment. To carry out this experiment we trained AdaBoost using the training set from the office environment. We then classified the test set from the corridor scenario. Table 5 shows the results of this classification. As expected, the errors increase compared to the situation in which the training and the test data were from the same domain. Even in this case, the classification rates remain above 90%. This indicates that our algorithm yields good generalizations and can also be employed for people detection in new environments.

2.3.3 Comparison with a Heuristic Approach

To analyze how much can be gained by our learning approach we compared the classification results of our AdaBoost-based classifier with the results obtained using a manually designed classifier that employs features frequently found in the literature

Table 2 Confusion matrix for the corridor environment

	Detected Label		
True Label	Person	No Person	Total
Person	**239** (99.58%)	**1** (0.42%)	**240**
No Person	**27** (1.03%)	**2589** (98.97%)	**2616**

Table 3 Confusion matrix for the office environment

	Detected Label		
True Label	Person	No Person	Total
Person	**497** (97.45%)	**13** (2.55%)	**510**
No Person	**171** (2.73%)	**6073** (96.26%)	**6244**

about laser-based people tracking. This classifier uses the following list of features and thresholds:

- Jump distance between adjacent beams for local minima extraction (features #4 and #5). The threshold for both features has been set to 30 cm.
- Segment width (feature #6). The corresponding thresholds derive from the task: Local minima blobs greater than 5 cm and smaller than 50 cm are accepted.
- Minimum number of points (feature #1). Segment with fewer than four points are discarded.
- Motion of beams (feature #14). Two consecutive scans are aligned and beam-wise subtracted from each other. Segments that contain beams which moved more than a certain distance are classified as people. This minimal distance was set to 2 cm, close above sensor noise.
- Standard deviation as a compactness measure of a segment (feature #2). The threshold was experimentally determined and set to 0.5 meter.

For this experiment we use the test set of the experiment explained is Section 2.3.1 where segments from the corridor and office were used together as examples. The results of the classification are shown in Table 7. As the results indicate, our approach yields much better results than the heuristic approach.

2.3.4 Adding the Motion Feature

In the previous experiments, only the first thirteen geometrical features were used. We additionally performed experiments after we added the motion feature #14. All scans from the corridor and the office runs were used for training and classification. The results are contained in Table 8. As can be seen, there is only a marginal improvement over the classifier without the motion feature (Table 4). Although the motion feature receives relatively high weight (it is ranked as the third most informative feature), this marginal improvement is simply an expression of the fact that people do not always move. People should be and – as this experiment demonstrates – can be detected without the use of motion information.

Table 4 Confusion matrix for both environments

	Detected Label		
True Label	Person	No Person	Total
Person	**722** (96.27%)	**28** (3.73%)	**750**
No Person	**225** (2.54%)	**8649** (99.88%)	**8860**

Table 5 Results obtained in the corridor environment using the classifier learned in the office

	Detected Label		
True Label	Person	No Person	Total
Person	**217** (90.42%)	**23** (9.58%)	**240**
No Person	**112** (4.28%)	**2504** (95.72%)	**2616**

Table 6 The best five features for each classifier

Environment	Five Best Features
Corridor	9, 4, 5, 2, 4
Office	9, 13, 3, 4, 5
Both	9, 13, 4, 3, 5

Table 7 Comparison with the heuristic approach

	Heuristic Approach	AdaBoost
False Negatives (%)	34.67	3.73
False Positives (%)	9.06	2.54
Overall Error (%)	11.06	2.63

Table 8 Classification errors after adding the motion feature

	Without Motion Feature	With Motion Feature
False Negatives (%)	3.73	3.47
False Positives (%)	2.54	3.13
Total Error (%)	2.63	3.15

2.3.5 Best Features for People Detection

As we use AdaBoost here as an offline method, that is, a technique that is run once and not on-the-fly on the robot, the question is: What are the best features for people detection in range data? The answer can be obtained from the importance of the individual feature weights in the final strong classifier. Table 6 lists the five best features for each AdaBoost classifier trained in the corridor, office and both environments respectively. Note that sometimes the same features occurs more than once in a classifier. In this case, they differ in their threshold and/or weight values.

It is interesting to interpret this outcome: The most informative feature is the radius of the circle fitted into the segment (feature #9). Note that this feature does not measure the degree of circularity (as feature #8) but is an alternative size estimate, apparently better than feature #6 (width). The mean angular difference (feature #3) is the second most important feature and quantifies the convexity of the segment. It is followed by the two jump distances (features #4 and #5) that we already know as the most popular detection features for local minima. Finally there are features #2 and #3 that both measure the compactness of the segment where feature #3 seems to be preferred. The reason for this is likely to be the more robust properties of the mean absolute deviation from the median over the simple standard deviation.

2.4 Summary

In this section, we addressed the problem of detecting people in laser range data. In contrast to previous approaches, which typically used manually tuned features and

thresholds, our approach applies the AdaBoost algorithm to learn a robust classifier. This classifier is created from a set of simple features that encode the geometry and statistics of a group of laser points and predicts if the points correspond to legs of people. The method has been implemented and applied in cluttered office environments. In practical experiments carried out in different environments we obtained encouraging detection rates of over 90%.

From the features selected by AdaBoost we can conclude that the shape of people in range data is best recognized by a radius feature, a convexity feature, a local minimum feature and a robust compactness feature (see Table 6).

3 Leg-Based People Tracking

Given the detection of legs of people in range data as presented in the previous section, we now address the tracking problem. In most related work on laser-based people tracking [23, 6, 7, 9, 10, 11, 24, 25, 26], a person is represented as a single state that encodes torso position and velocities. Clearly, the appearance of people in laser range data depends on the mounting height of the sensor: at hip height a human torso is typically seen as a single local-minimum blob, while at foot height, legs produce separate, fast-moving smaller blobs. In practice, the mounting height of the sensor is often constrained by the application or the robot's form factor and not only by the researcher to suit the needs of a tracking algorithm. Safety regulations, for instance, require laser scanners to be mounted at foot height. At this height, single blobs are poor models for the appearance of people which motivates leg tracking as an approach to laser-based people tracking. Accordingly, the problem of people tracking has recently been addressed as a leg tracking problem [27, 28] where people are represented by the states of two legs, either in a single augmented state [28] or as a high-level track to which two low-level leg tracks are associated [27].

Multi-hypothesis tracking (MHT) [3, 29] belongs to the most general data association techniques as it produces joint compatible assignments, integrates them over time, and is able to deal with track creation, confirmation, occlusion, and deletion. Other multi-target data association techniques such as the nearest neighbor filter, the track splitting filter, or the JPDAF are less powerful or sub-optimal in nature [30].

In the context of people tracking with laser range finders, Taylor *et al.* [27] employ an MHT to resolve ambiguities in the problem of fitting a walking person into two leg measurements. The authors use a geometric occlusion model to decrease the detection probability if an occlusion is to be expected. Mucientes *et al.* [26] cluster people into groups and utilize an MHT to handle the assignments of measurements to single tracks and clusters. Given the high-level concept of groups, additional assignments of measurements to tracks within groups become possible for which the authors derive appropriate probabilities.

As mentioned before, the people tracking technique described here builds upon the leg detection described in Section 2. The proposed approach tracks legs of people and utilizes a multiple hypothesis tracking technique for data association. Opposed to most related work in the laser-based people tracking literature, we address

the problem of tracking legs that are measured individually. Based on the resulting leg tracks, we create person tracks using the multivariate weighted mean if two tracks are sufficiently close and move in the same direction for a certain time frame. Once a person track has been validated over time, we adapt the individual occlusion probabilities of both associated leg tracks to account for the fact that legs frequently occlude each other. To this end, we extend the MHT framework to explicitly take into account potential occlusions by introducing adaptive conditional assignment probabilities.

3.1 The KF-Based Tracker

This section describes the KF-based multi-target tracker that is used to track legs of people. We briefly go through the tracking cycle. For the details of Kalman filtering and target tracking the reader is referred to Bar-Shalom and Li [30].

State prediction. A leg track is represented as $\mathbf{x} = (x, y, v_x, v_y)$ where x and y are the track position and v_x and v_y the x and y components of the track velocity. With this state representation new tracks can be properly initialized with $v_x = v_y = 0$. For motion prediction, a constant velocity model is employed.

Measurement prediction. As the x- and y-coordinates of a track are directly observable, the 2×4 measurement matrix H is formed by the 2×2 identity matrix in x and y and the 2×2 zero matrix in v_x and v_y.

Observation. The observation step consists in detecting people in range data. The problem can be seen as a classification problem that consists in finding those laser beams that correspond to people and to discard other beams. Typically, hand designed classifiers have been employed for this task with a manual selection of features and thresholds. In this work, we use the approach described in Section 2 to detect legs from laser data.

The observation step delivers the set of observations (or measurements) $\mathbf{z}_k = \{z_k^1, z_k^2, \ldots, z_k^{M_k}\}$ at time index k. M_k denotes the current number of measurements.

Data association. For data association we employ a modified MHT approach described in the sections hereafter.

Estimation. Given that both, the state and measurement prediction models are linear, a (non-extended) Kalman filter as the optimal estimator under the Gaussian assumption can be employed.

3.2 The Multi-Hypothesis Leg Tracker

In this section we review the MHT as described in the two papers by Reid [3] and Cox *et al.* [29]. In summary, the MHT hypothesizes about the state of the world by considering all statistically feasible assignments between measurements and tracks and all possible interpretations of measurements as false alarms or new tracks and tracks as matched, occluded or obsolete. A hypothesis Ω_j^k is one possible set of assignments and interpretations at time k.

In the original paper by Reid [3] measurements are interpreted as matches with existing tracks, new tracks, or false alarms. Tracks are interpreted as *detected* (when

Table 9 Example of an assignment.

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_{nt}$	$\mathbf{x}_{fa}$
z_1	0	0	1	0
z_2	1	0	0	0
z_{del}	0	1	0	0

they match with a measurement) or *not detected*. Deletions of tracks are not handled by the MHT but by a heuristics based on sequences of consecutive non-detections. Cox *et al.* [29] extend this framework with the interpretation of tracks as *deleted*. Thereby, the MHT handles the entire life-cycle of tracks from creation over matching to termination. Occlusions are considered as simultaneous non-detection and non-deletion events.

In order to *adapt* the occlusion probability of individual leg tracks, it is necessary to reconsider the derivation of the hypothesis probabilities in the MHT, especially the assignment set probabilities. Let Ω_j^k be the j-th hypothesis at time k and $\Omega_{p(j)}^{k-1}$ the parent hypothesis from which Ω_j^k was derived. Let further $\Psi_j(k)$ denote a set of assignments that, based on the parent hypothesis $\Omega_{p(j)}^{k-1}$ and the current measurement $\mathbf{z}_k$, gives rise to Ω_j^k.

The assignment set $\Psi_j(k)$ associates each measurement either to an existing track, a false alarm, or a new track and marks a track as *detected* or *deleted*. Assignment sets are best visualized in matrix form such as the example in Table 9 that shows a set of assignments of tracks $\mathbf{x}_1, \mathbf{x}_2$ with measurements z_1 and z_2. An assignment is denoted by a non-zero entry in the matrix. The example shows a situation in which track $\mathbf{x}_1$ is assigned to measurement z_2, track $\mathbf{x}_2$ is scheduled for deletion, and measurement z_1 is interpreted as a new track. There are as many possible assignment sets $\Psi_j(k)$ as we can distribute 1's and 0's over such matrices under the constraints of unique measurement-to-track associations and that the only zero-valued rows and columns can belong to the events *deletion*, *new track*, and *false alarm*. An assignment set has a probability that is determined by the probabilities of these events and the probability of a specific distribution of 1's and 0's.

Given an assignment set probability and the probability of the parent hypothesis $\Omega_{p(j)}^{k-1}$, we can calculate the probability of each child hypothesis that has been created as $\Psi_j(k)$. This calculation is done recursively [3]:

$$\begin{aligned} p(\Omega_j^k|\mathbf{z}_k) &= p(\Psi_j(k), \Omega_{p(j)}^{k-1}|\mathbf{z}_k) \\ &\overset{\substack{Bayes+\\ Markov}}{=} \eta\, p(\mathbf{z}_k|\Psi_j(k), \Omega_{p(j)}^{k-1}) p(\Psi_j(k)|\Omega_{p(j)}^{k-1}) \cdot p(\Omega_{p(j)}^{k-1}). \end{aligned} \tag{2}$$

The rightmost term on the right-hand side is the recursive term, that is, the probability of its parent. Factor η is a normalizer. The leftmost term on the right-hand side after the normalizer η is the measurement likelihood. We assume that a measurement z_k^i associated to a track $\mathbf{x}_j$ has a Gaussian pdf centered around the measurement prediction $\hat{z}_k^j$ with innovation covariance matrix $\mathcal{S}_k^{i,j}$, $\mathcal{N}(z_k^i) := \mathcal{N}(z_k^i; \hat{z}_k^j, \mathcal{S}_k^{i,j})$. We

further assume the pdf of a measurement belonging to a new track or false alarm being uniform in the observation volume V (the field of view of the sensor) with probability V^{-1}. Thus

$$p(\mathbf{z}_k|\Psi_j(k),\Omega_{p(j)}^{k-1}) = \prod_{i=1}^{M_k}\mathcal{N}(z_k^i)^{\delta_i}V^{1-\delta_i}$$
$$= V^{-(N_{fal}+N_{new})}\prod_{i=1}^{M_k}\mathcal{N}(z_k^i)^{\delta_i} \quad (3)$$

with N_{fal} and N_{new} the number of measurements labeled as *false alarms* and *new tracks* respectively. δ_i is an indicator variable being 1 if and only if measurement i has been associated to a track, 0 otherwise.

The central term on the right-hand side of Equation (2) is the probability of an assignment set, $p(\Psi_j(k)|\Omega_{p(j)}^{k-1})$, which is composed of three terms:

1. The probability of the *number* of tracks N_{det}, N_{fal}, N_{new} with a certain label. In Reid's case, with tracks being either labeled *detected* or *not detected*, the number of detected tracks N_{det} given the total number of tracks in the parent hypothesis, N, follows a binomial distribution

$$p(N_{det}|\Omega_{p(j)}^{k-1}) = \binom{N}{N_{det}} p_{det}^{N_{det}}(1-p_{det})^{(N-N_{det})} \quad (4)$$

Assuming that the *number* of false alarm and the *number* of new tracks both follow a Poisson distribution with expected number of events $\lambda_{fal}V$ and $\lambda_{new}V$ in the observation volume V respectively, we obtain

$$p(N_{det},N_{fal},N_{new}|\Omega_{p(j)}^{k-1}) =$$
$$\binom{N}{N_{det}} p_{det}^{N_{det}}(1-p_{det})^{(N-N_{det})}\cdot\mu(N_{new};\lambda_{new}V)\cdot\mu(N_{fal};\lambda_{fal}V) \quad (5)$$

where $\mu(n;\lambda V)$ is the Poisson distribution for n events when the average rate of events is λV.

2. The probability of a specific assignment of measurements so that $M_k = N_{det} + N_{fal} + N_{new}$ holds. The probability is determined as 1 over the number of combinations which is

$$\binom{M_k}{N_{det}}\binom{M_k-N_{det}}{N_{fal}}\binom{M_k-N_{det}-N_{fal}}{N_{new}} \quad (6)$$

where the last term equals 1.

3. The probability of a specific assignment of tracks given that a track can either be *detected* or *not detected*. The probability is determined as 1 over the number of these assignments

$$\frac{N!}{(N-N_{det})!}\binom{N-N_{det}}{N_{det}}. \quad (7)$$

The first term follows from the combinatorial fact, that a track can be chosen only once and the track-to-measurement order matters.

It is noteworthy (and one of the key contributions of Reid [3]) that in the product of these three probabilities many terms cancel out, and substituted into the Equation (2), the final probability $p(\Omega_j^k|\mathbf{z}_k)$ becomes a simple and easy to calculate expression independent of the observation volume V.

3.3 Person Tracking and Occlusion Adaptation

The tracking system presented in the previous sections maintains N tracks that correspond to human legs. Only on the level of these N tracks, we reason on the existence of people by the use of the following model knowledge:

1. People have always two legs
2. Legs are close to each other
3. Legs move in a similar direction
4. Legs have a higher probability of occluding each other than being occluded by other people's legs or objects

In contrast to previous works that consider gait models of walking persons [27, 28] we deliberately take a minimalist approach since people have a large variety of leg geometries and motion patterns, poorly captured by a gait model. Thus, to create a person track, we implement each point of the above-mentioned model as follows:

1. A person track is defined as a high-level track to which two legs tracks are associated. The state of a person is estimated from the state of the two legs tracks using the multivariate weighted mean.
2. Two tracks $\mathbf{x}_i, \mathbf{x}_i$ that satisfy a nearness condition given a threshold θ_d which in our case is set to 0.75 meter form a person candidate.
3. A person candidate is validated if the two tracks maximize the scalar product of their orientations summed over the track histories $S = \sum_t < \theta_i^t, \theta_j^t >$ with $\theta_i = \text{atan2}(v_{y,i}^2, v_{x,i}^2)$ being the orientation of track $\mathbf{x}_i$. In practice, we calculate S only in a sliding window over the last L steps and validate a person track that satisfies $S > \theta_a$ where θ_a is an experimentally determined threshold.
4. The adaptation of the occlusion probability is described in detail in the following subsection.

Person tracks are estimated based on its associated leg tracks. Thus, a person track is deleted either if the MHT deletes one or both of its leg tracks or if condition 2 does not hold anymore for L consecutive steps.

3.3.1 Adaptation of Occlusion Probability

According to Reid [3], who only considers the label *detected*, the number of tracks with this label, N_{det}, follows a binomial distribution. In the more general case, in

which we have an arbitrary number of labels, the number of tracks with a given label follows a *multinomial distribution*.

Besides *detection* (according to Reid [3]) and *deletion* (introduced by Cox and Hingorani [29]) we introduce the label *occlusion*. Thus, the pdf of the labeling of the tracks into *detected*, *occluded*, and *deleted* is

$$p(N_{det}, N_{occ}, N_{del} | \Omega_{p(j)}^{k-1}) = \frac{N!}{N_{det}! N_{occ}! N_{del}!} p_{det}^{N_{det}} p_{occ}^{N_{occ}} p_{del}^{N_{del}} \tag{8}$$

with $p_{det} + p_{occ} + p_{del} = 1$ and $N = N_{det} + N_{occ} + N_{del}$. Equation (8) is the generalization of Equation (4) and allows to specifically adjust the label probabilities. Occlusions are no longer implied by non-detection and non-deletion but are made explicit as a label with their own specific probability.

However, adjusting individual probabilities raises the question whether probabilities of assignments and hypotheses remain properly normalized across branches in the hypothesis tree. We will now verify that the consistency in this sense is maintained.

In our case, there are leg tracks that are associated to validated person tracks and leg tracks that are either associated to non-validated person tracks or to no person track at all. We will denote the former as *approved* (by the superscript A) and the latter as *free* (by the superscript F). With N^A and N^F as the number of approved and the number of free tracks respectively, $N = N^A + N^F$ and likewise

$$N^F = N_{det}^F + N_{occ}^F + N_{del}^F \tag{9}$$

$$N^A = N_{det}^A + N_{occ}^A + N_{del}^A. \tag{10}$$

The evidence *approved* and *free* conditions the probabilities in Equation (8) such that the right-hand side must be rewritten as the product of two multinomial distributions, each with three conditional probabilities $p_{det|F}$, $p_{del|F}$, $p_{occ|F}$ and $p_{det|A}$, $p_{del|A}$, $p_{occ|A}$ for which $p_{det|F} + p_{del|F} + p_{occ|F} = 1$ and $p_{det|A} + p_{del|A} + p_{occ|A} = 1$ must hold. The product of multinomial distributions is explained by the fact that a track can only be either approved or free.

As a consequence, the three product terms that compose the assignment set probability, $p(\Psi_j(k) | \Omega_{p(j)}^{k-1})$, are altered as follows. The first term, the probability of the *number* of tracks with a certain label becomes

$$\begin{aligned} &p(N_{det}^F, N_{occ}^F, N_{del}^F, N_{det}^A, N_{occ}^A, N_{del}^A, N_{new}, N_{fal} | \Omega_{p(j)}^{k-1}) \\ &= \frac{N^F!}{N_{det}^F! N_{occ}^F! N_{del}^F!} \cdot p_{det|F}^{N_{det}^F} \cdot p_{occ|F}^{N_{occ}^F} \cdot p_{del|F}^{N_{del}^F} \cdot \\ &\quad \frac{N^A!}{N_{det}^A! N_{occ}^A! N_{del}^A!} \cdot p_{det|A}^{N_{det}^A} \cdot p_{occ|A}^{N_{occ}^A} \cdot p_{del|A}^{N_{del}^A} \cdot \\ &\quad \mu(N_{fal}; \lambda_{fal} V) \cdot \mu(N_{new}; \lambda_{new} V) \end{aligned} \tag{11}$$

The second term, the probability of a specific combination of these numbers, is calculated as 1 over the number of these combinations, which is

$$\begin{pmatrix} M_k \\ N_{det}^F \end{pmatrix} \begin{pmatrix} M_k - N_{det}^F \\ N_{det}^A \end{pmatrix} \begin{pmatrix} M_k - N_{det}^F - N_{det}^A \\ N_{new} \end{pmatrix} \cdot \begin{pmatrix} M_k - N_{det}^F - N_{det}^A - N_{new} \\ N_{fal} \end{pmatrix} = \frac{M_k!}{N_{det}^F! N_{det}^A! N_{new}! N_{fal}!} \tag{12}$$

since $M_k = N_{det}^F + N_{det}^A + N_{new} + N_{fal}$.

Similarly, for the third term, the probability of the number of track-to-measurement associations determined as 1 over the number of these associations, is 1 over

$$\frac{N^F!}{(N^F - N_{det}^F)!} \begin{pmatrix} N^F - N_{det}^F \\ N_{occ}^F \end{pmatrix} \begin{pmatrix} N^F - N_{det}^F - N_{occ}^F \\ N_{del}^F \end{pmatrix} \cdot \frac{N^A!}{(N^A - N_{det}^A)!} \begin{pmatrix} N^A - N_{det}^A \\ N_{occ}^A \end{pmatrix} \begin{pmatrix} N^A - N_{det}^A - N_{occ}^A \\ N_{del}^A \end{pmatrix} = \frac{N^F! N^A!}{N_{occ}^F! N_{del!}^F N_{occ}^A! N_{del}^A!} \tag{13}$$

When combining these results, many terms cancel out like in Reid's approach [3]. Accordingly, we obtain the assignment set probability as

$$p(\Psi_j(k)|\Omega_{p(j)}^{k-1}) = \eta' \cdot p_{det|F}^{N_{det}^F} \cdot p_{occ|F}^{N_{occ}^F} \cdot p_{del|F}^{N_{del}^F} \cdot p_{det|A}^{N_{det}^A} \cdot p_{occ|A}^{N_{occ}^A} \cdot p_{del|A}^{N_{del}^A} \cdot \lambda_{new}^{N_{new}} \cdot \lambda_{fal}^{N_{fal}} \cdot V^{N_{new}+N_{fal}} \tag{14}$$

where η' is a constant normalization factor.

Substituting Equation (14) and the measurement likelihood from Equation (3) into Equation (2) yields the final expression for the probability of a child hypothesis

$$p(\Omega_j^k|\mathbf{z}_k) = \eta'' \prod_{i=1}^{M_k} \mathcal{N}(z_k^i)^{\delta_i} \cdot p_{det|F}^{N_{det}^F} \cdot p_{occ|F}^{N_{occ}^F} \cdot p_{del|F}^{N_{del}^F} \cdot p_{det|A}^{N_{det}^A} \cdot p_{occ|A}^{N_{occ}^A} \cdot p_{del|A}^{N_{del}^A} \cdot \lambda_{new}^{N_{new}} \cdot \lambda_{fal}^{N_{fal}} \cdot p(\Omega_{p(j)}^{k-1}). \tag{15}$$

Here $\eta'' = \eta \cdot \eta'$ is a constant normalization factor which ensures that the probabilities of the hypotheses Ω_j^k sum up to 1. It can be shown that η'' only depends on M_k. This means that within the same generation of hypotheses – for which M_k is identical – proper normalization across all branches in the tree, that is across all hypothesis probabilities, is guaranteed.

3.3.2 Branching and Pruning Strategies

For an efficient implementation of an MHT, pruning strategies that limit the exponential explosion of hypotheses are mandatory. As proposed by Cox and Hingorani [29] we make use of the following strategies:

- *k-Best Branching*. Instead of creating all children, we generate only the k best children for each parent hypothesis. This can be done in polynomial time with an algorithm proposed by Murty [31].
- *N-scan-back*. The N-scan-back algorithm considers an ancestor hypothesis at time $k-N$ and looks ahead in time to all its children at the current time k (the leaf nodes). It evaluates the probabilities of all leaf nodes, keeps the best branch at time $k-N$ in terms of probability mass and discards all others.

3.4 Experimental Evaluation

The approach described above has been implemented and evaluated on a an ActiveMedia Powerbot mobile robot equipped with a Sick LMS laser scanner mounted at a height of 11 cm above ground. The angular resolution of the range scans was 0.5°. Throughout all experiments we used the values listed in Table 10 for the conditional probabilities introduced in the previous section. These values can be learned from a labeled test data set. Our adaptive method uses the probabilities with the superscript F for *free* tracks and the probabilities with the superscript A for *approved* tracks. We compare our method also to the non-adaptive case for which we use the probabilities with the superscript F as default values unless otherwise noted.

3.4.1 Person Walking on an 8-Shaped Trajectory

In the first experiment a person follows a 8-shaped trajectory in a corridor of about 2.5 meters width in normal walking speed. As can be seen from Figure 5, our system was able to reliably track the person despite the fact that it only used a constant velocity motion model to track the sharp turns carried out by the person. The same leg tracks last over the entire duration of the experiment. This is illustrated by the diagram in right image of Figure 5 that shows a constant number of four tracks. Two of the four tracks are due to false alarms extracted in the clutter. Without adaptation of the occlusion probability, there is track loss at nearly every U-turn giving rise to many newly created tracks.

Table 10 Parameters used throughout the experiments.

$p_{det\|F}$	$p_{occ\|F}$	$p_{del\|F}$	$p_{det\|A}$	$p_{occ\|A}$	$p_{del\|A}$	λ_{new}	λ_{fal}
0.3	0.63	0.07	0.2	0.79	0.01	0.001	0.003

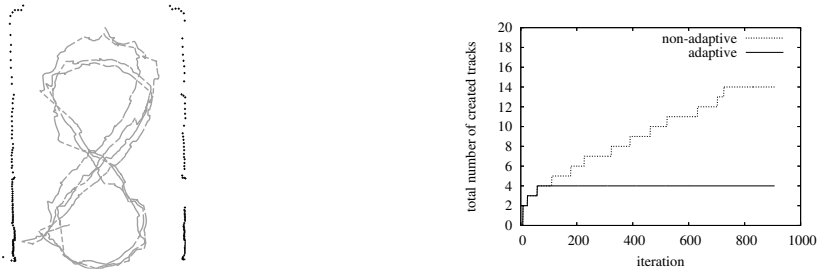

Fig. 5 Trajectories and total numbers of created tracks for experiment 1.

3.4.2 Person Turning Constantly while Moving Forward

In the second experiment a person is moving on a straight line turning 180° around the stationary leg at each step (see Figure 6). This unusual walking pattern produces heavy occlusions of the moving leg by the stationary one. The adaptive approach was able to track the person accurately during the experiment. The total number of tracks in Figure 6 (right) is constant (three), one of them being a false alarm. The mutual leg occlusion is poorly handled by the non-adaptive approach as the increasing number of new tracks in the diagram illustrates.

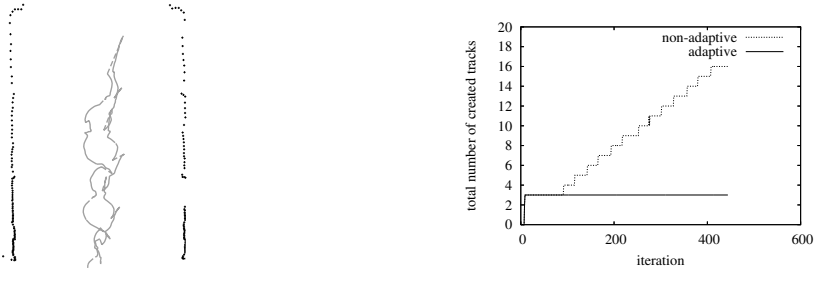

Fig. 6 Trajectories and total numbers of created tracks for experiment 2.

3.4.3 People Walking Randomly in a Narrow Corridor

It remains to be demonstrated that the superior performance of the adaptive approach found so far is not just due to better tuned probability parameters for approved tracks. This is demonstrated in the third experiment where up to four people simultaneously move through the field of view of the sensor. The subjects perform typical motion patterns at normal walking speed, they avoid each other, turn on the spot, cross paths, stop once in a while, and frequently enter and leave the field of view. This leads up to four validated people tracks simultaneously (eight leg tracks), not included false alarms due to, e.g., corners falsely detected as legs.

Figure 7 shows a portion of the experiment with four simultaneously tracked people. The chance of additional mutual occlusions from people is substantial in

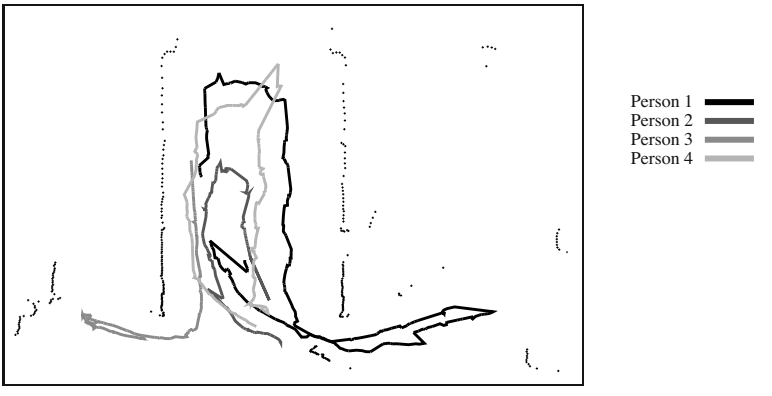

Fig. 7 Trajectories of four people tracks during experiment 3.

this narrow environment. Figure 8 depicts the total number of created tracks. Due to long lasting occlusions produced by other people, the system sometimes deletes tracks although the person is still there, and creates new tracks when the person becomes visible again. However, Figure 8 shows that compared to the non-adaptive case, we are able to track people more robustly over an extended period of time as the number of tracks is substantially closer to ground truth. The ground truth information was obtained by manual inspection.

If we use the parameter setting for approved tracks as default (and without adaptation), we observe in Figure 9 (left) that the number of simultaneous tracks nearly never decreases, that is, tracks are deleted with a very low probability. When tracks are not deleted, their uncertainty grows boundless producing a high level of ambiguity, and ergo, a high number of matching candidates that pass the Mahalanobis test. This causes an explosion of branches in the hypothesis tree as illustrated in Figure 9 (right). The diagram shows the number of hypotheses between steps 900 and 1000, the time when all four people were in the field of view. In the adaptive case, the peak

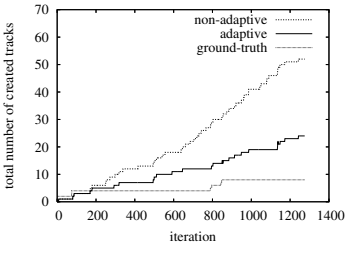

Fig. 8 Total number of created tracks for the adaptive method, the non-adaptive method, and the ground-truth.

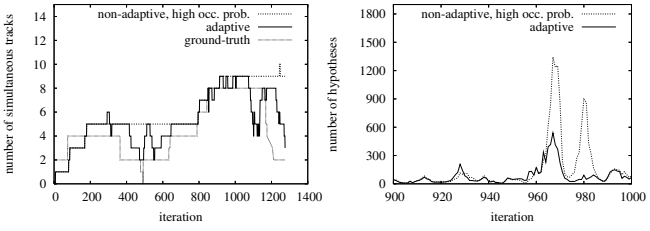

Fig. 9 Number of tracks for the adaptive method, the non-adaptive method with parameters for approved tracks as default versus the ground truth (left) and number of simultaneous hypotheses for our adaptive case and the non-adaptive method with parameters for approved tracks as default (right).

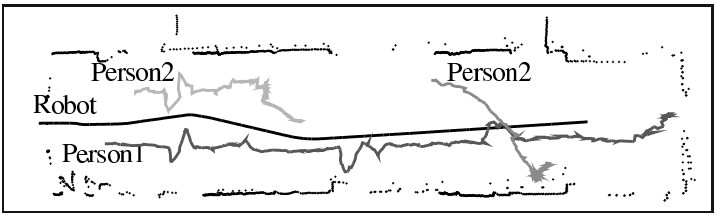

Fig. 10 Trajectories of robot and people in experiment 4. Person 1 is constantly tracked, person 2 receives a new identifier when reentering the sensor's field of view.

numbers of hypotheses are seriously more moderate compared to the non-adaptive approach where the parameters for approved tracks are taken as default.

The average cycle time in this experiment with four people was 44.5 ms on an Athlon 4400+ and with a scan-back depth of eight (see section 3.3.2). A significant acceleration (from initially 220 ms) was due to the introduction of separate trees for tracks and hypotheses as proposed by Cox and Hingorani [29] that avoids processing duplicate tracks.

3.4.4 Tracking from a Moving Robot

In the forth experiment the robot moves with an average translational velocity of 0.33 m/s (max. 0.5 m/s) while tracking two people (Figure 10). The two subjects move at normal walking speeds, stop once in a while with person 2 leaving and re-entering the robot's field of view. Consecutive scans are aligned using odometry information. With a moving sensor, detection of moving leg blobs is more difficult as also the background becomes dynamic. Especially in clutter the AdaBoost classifier therefore generates a higher number of false alarms. Because people tracks are initialized only from leg tracks that satisfy our person model, the robot is able to robustly track the two people with only one incorrect people track that appears for two iterations. The non-adaptive approach creates additionally eleven incorrect leg tracks resulting in a total of four incorrect people tracks.

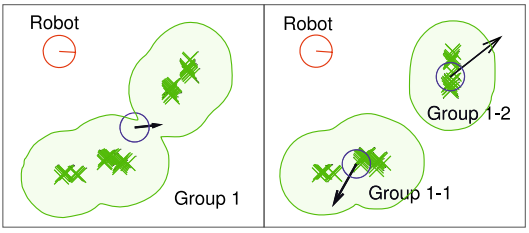

Fig. 11 Tracking groups of people with a mobile robot. Groups are shown by their position (blue), velocity (black), the associated laser points (green), and a contour for visualization. In the two frames, a group of four people splits up into two groups with two people each.

3.5 Summary

Based on the leg detection method presented in Section 2, this section presented our approach to people tracking posed as a leg tracking problem. For data association, we utilized an MHT that we extended to incorporate adaptive occlusion probabilities and present a mathematical derivation for this approach. The approach has been implemented and tested on a real robot with data acquired by a SICK laser range sensor. The experimental results demonstrate that our approach is able to robustly track multiple people based on observations of their legs even when enduring occlusions occur. We also carried out experiments that demonstrate that our adaptive approach outperforms a non-adaptive MHT with fixed occlusion probability settings, since the latter overly delays track deletion and thus produces a high level of ambiguity coupled with an explosion of the number of hypotheses. Our system is able to perform each update fast enough for online processing on a standard PC even when the robot is tracking four people.

4 Tracking Groups of People

The people tracking approach presented so far can robustly detect and track individuals. For crowded environments, however, tracking single people can not only become increasingly difficult but also ignores an important human trait namely that people are social beings. As such they form groups, interact with each other, merge to larger groups, or separate from groups. Tracking individual people in these formation processes can be hard due to the high chance of occlusion and the large extent of data association ambiguity. This causes the space of possible associations to become huge and the number of assignment histories to quickly become intractable. Further, for many applications, knowledge about groups can be sufficient as the task does not require to know the state of every person. In such situations, tracking groups that consist of multiple people is more efficient. Additionally, it reveals semantic information about activities and social relations of people.

This section focuses on group tracking in populated environments with the goal to track a large number of people in real-time. The approach attempts to maintain

the state of groups of people over time, considering possible splits and merges as shown in Fig. 11. Again, for our experiments we use a mobile robot equipped with a laser range finder, but our method is applicable to data from other sensors as well.

In most of the related work on laser-based people tracking, tracks correspond to individual people [23, 6, 9, 11, 24]. In Taylor *et al.* [27] as well as in the approach described in the previous section, tracks represent the state of legs which are fused to people tracks in a later stage.

Khan *et al.* [32] proposed an MCMC-based tracker that is able to deal with non-unique assignments, i.e., measurements that originate from multiple tracks, and multiple measurements that originate from the same track. Actual tracking of groups using laser range data was, to our knowledge, first addressed by Mucientes *et al.* [26]. Most research in group tracking was carried out in the vision community [33, 34, 35]. Gennari *et al.* [34] and Bose *et al.* [35] both address the problem of target fragmentation (splits) and grouping (merges). They do not integrate data association decisions over time – a key property of the Multi-Hypothesis Tracking (MHT) approach, initially presented by Reid [3] and later extended by Cox *et al.* [29]. The approach belongs to the most general data association techniques as it produces joint compatible assignments, integrates them over time, and is able to deal with track creation, matching, occlusion, and deletion. association techniques such as the nearest neighbor filter, the track splitting filter, or the JPDAF are less powerful or sub-optimal in nature.

The works closest to our approach are Mucientes *et al.* [26] and Joo *et al.* [36]. Both address the problem of group tracking using an MHT approach. Mucientes *et al.* employ two separate MHTs, one for the regular association problem between observations and tracks, and a second stage MHT that hypothesizes over group merges. However, people tracks are not replaced by group tracks, hence there is no gain in efficiency. The main benefit of that approach is the additional semantic information about the formation of groups.

Joo *et al.* [36] present a vision-based group tracker using a single MHT to create hypotheses of group splits and merges and observation-to-track assignments. They develop a variant of Murty's algorithm [31] that generates the k-best *non-unique* assignments which enables them to make multiple assignments between observations and tracks, thereby describing target splits and merges. However, the method only produces an approximation of the optimal k-best solutions since the posterior hypothesis probabilities depend on the number of splits, which, at the time when the k-best assignments are being generated, is unknown. In our approach, the split, merge and continuation events are given by the model *before* computing the assignment probabilities, and therefore, our k-best solutions are optimal.

In this section we propose a tracking system for groups of people using an extended MHT approach to hypothesize over both, the group formation process (models) and the association of observations to tracks (assignments). Each model, defined to be a particular partitioning of tracks into groups, creates a new tree branch with its own assignment problem. As a further contribution we propose a group representation that includes the shape of the group, and we show how this representation is updated in each step of the tracking cycle. This extends previous approaches to

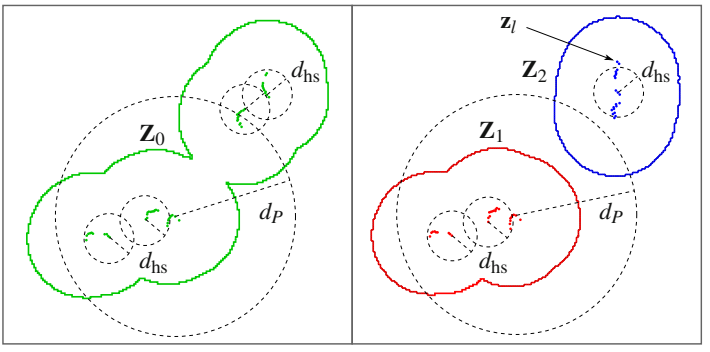

Fig. 12 Illustration of the detection step. *Left:* One group is detected since all shortest links between the measured points z_l are smaller than the single-linkage clustering threshold d_P. *Right:* Two groups are found as the shortest link between their points exceeds d_P. For group size estimation, the number of human-sized blobs in a group is determined by applying the same clustering procedure with threshold d_{hs}.

group tracking where groups are assumed to have Gaussian shapes only [34, 26]. Our proposed group tracker also estimates the number of people in groups and employs a labeling system to represent the history of group interactions, both of which extend the approach presented in our previous work [37].

Finally, we use the psychologically motivated *proxemics* theory introduced by Hall [4] for the definition of a group. The theory relates social relation and body spacing during social interaction and proposes thresholds that separate the intimate, personal, social, and public space around people.

4.1 Group Detection in Range Data

Detecting people in range data has been approached with motion and shape features [23, 6, 9, 11, 24, 26] as well as with the classifier described in Section 2. However, these systems were designed (or trained) to extract single people. In the case of densely populated environments, groups of people typically produce large blobs in which individuals are hard to recognize. We therefore pursue the approach of background subtraction and clustering. Given a previously learned model (a map of the environment for mobile platforms), the background is subtracted from the scans and the remaining points are passed to the clustering algorithm. This approach is also able to detect standing people as opposed to the work of Mucientes *et al.* [26] which relies on motion features. Note that the detection method is not critical to the system and could also be replaced by map-free approaches that employ appearance information, motion features, or other filtering techniques.

Concretely, a laser scanner generates measurements consisting of bearing and range values. The measurements are transformed into Cartesian coordinates $\mathbf{z}_l = (x_l, y_l)^T$ and grouped using *single linkage clustering* [38] with a distance threshold d_P. The outcome is a set of clusters $\mathcal{Z}_i$ making up the current observation set

$Z(k) = \{\mathcal{Z}_i \,|\, i = 1, \ldots, N_{\mathcal{Z}}\}$, where each cluster represents a single observation in the tracking framework. Each cluster $\mathcal{Z}_i$ is a complete set of measurements $\mathbf{z}_l$ that fulfills the cluster condition, i.e., two clusters are joined if the distance between their closest points is smaller than d_P. A similar concept, using a connected components formulation, has been used by Gennari and Hager [34]. The clusters then contain range readings that can correspond to single legs, individual people, or groups of people, depending on the cluster distance d_P.

Even though tracking of individuals in groups is not feasible due to frequent occlusions, the number of detected individuals in a group correlates with the true number of people in a group. As an observation of the group size, we therefore take the number of human-sized clusters $n_{\text{hs}}(\mathcal{Z}_i)$ found in an observation cluster $\mathcal{Z}_i$. We determine this by counting the clusters after reapplying single linkage clustering to the points in $\mathcal{Z}_i$ with an appropriate distance threshold d_{hs}, with $d_{\text{hs}} < d_P$.

An example for the clustering is given in Fig. 12. On the left, all links are shorter than d_P so that the measurements are grouped into one cluster $\mathcal{Z}_0$ that contains four human-sized clusters. On the right, the shortest distance between the two groups exceeds d_P so that they are kept as two clusters, $\mathcal{Z}_1$ and $\mathcal{Z}_2$. The two people in $\mathcal{Z}_2$ are counted as only one human-sized cluster.

4.2 Group Definition and Group Tracks

This section defines the concept of a group, describes the initialization of group tracks and derives the probabilities of group-to-observation assignments and group-to-group assignments.

What makes a collection of people a *group* is a highly complex question in general, which involves social relations among subjects that are difficult to measure. A concept related to this question is the proxemics theory introduced by Hall [4] who found from a series of psychological experiments that social relations among people are reliably correlated with physical distance during interaction. This finding allows us to infer group affiliations by means of body spacing information available in the range data. The distance d_P thereby becomes a threshold with a meaning in the context of group formation.

4.2.1 Representation of Groups

Concretely we represent a group as a tuple $G = \langle \mathbf{x}, C, \mathcal{P}, \mathcal{L} \rangle$ with $\mathbf{x}$ as the track state, C the state covariance matrix, $\mathcal{P}$ the set of contour points that belong to G, and $\mathcal{L}$ the set of identification labels. The track state vector $\mathbf{x} = (x, y, \dot{x}, \dot{y}, n)^T$ is composed of the position $(x, y)^T$, the velocities $(\dot{x}, \dot{y})^T$, and n, the number of people in the group.

The points $\mathbf{x}_{\mathcal{P}_l} \in \mathcal{P}$ are an approximation of the current shape or spatial extension of the group. Shape information will be used for data association under the assumption of *instantaneous rigidity*. That is, a group is assumed to be a rigid object over the duration of a time step Δt, and consequently, all points in $\mathcal{P}$ move coherently with the estimated group state $\mathbf{x}$. The points $\mathbf{x}_{\mathcal{P}_l}$ are represented relative to the state $\mathbf{x}$, as described in Sect. 4.2.2.

The label set $\mathcal{L}$ contains identification labels that are associated with the group. These labels explicitly represent the history of track interactions, which can be of high interest for certain applications, e.g., to determine which people belong together.

4.2.2 Initialization of Group Tracks

If the tracker creates a new group track G_j from an observation cluster $\mathcal{Z}_i$ in time step k, the positional components $(x_j, y_j)^T$ of track state $\mathbf{x}_j(k|k)$ are initialized with the centroid position of the measurement cluster. The contour points $\mathcal{P}_j$ are the points in $\mathcal{Z}_i$ represented relative to the centroid (omitting the time index $(k|k)$ for readability):

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} := \bar{\mathbf{z}}_i = \frac{1}{|\mathcal{Z}_i|} \sum_{z_l \in \mathcal{Z}_i} z_l \,, \quad \mathcal{P}_j := \bigcup_{z_l \in \mathcal{Z}_i} z_l - \bar{\mathbf{z}}_i \,. \tag{16}$$

The unobserved velocity components $(\dot{x}_j, \dot{y}_j)^T$ of $\mathbf{x}$ are set to zero, the size estimate is set to the number of human-sized blobs in the measurement cluster, $n_j := n_{\mathrm{hs}}(\mathcal{Z}_i)$, and the label set is assigned a unique number as its only element, e.g., $\mathcal{L}_j := \{0\}$ for the first group after starting up the tracker. The initial state covariance is given by $C_j = C_0$, where C_0 is a diagonal matrix with $\left(\sigma_x{}^2, \sigma_y{}^2, \sigma_{\dot{x}}{}^2, \sigma_{\dot{y}}{}^2, \sigma_n{}^2\right)$ being the elements on the main diagonal. To account for the unknown components in the initial state vector, high uncertainty values are used for the corresponding entries in the initial state covariance matrix.

4.2.3 Motion Model for Group Tracks

To track groups over time, the state $x(k|k)$ and state covariance $C(k|k)$ of each group track in time step k are predicted into the next time step using a motion model. The predictions are denoted as $x(k+1|k)$ and $C(k+1|k)$, respectively. For tracks that are *continued*, i.e., no splits or merges take place from one frame to the next, we assume constant velocity for the centroid of the group, and a constant number of people in the group. Using a linear Kalman filter we get $\mathbf{x}(k+1|k) = A\,\mathbf{x}(k|k)$ and $C(k+1|k) = A\,C(k|k)A^T + Q$ for the state prediction. The state transition matrix A and the process noise covariance matrix Q are given by

$$A = \begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \; Q = \begin{pmatrix} \varepsilon_x{}^2 & 0 & 0 & 0 & 0 \\ 0 & \varepsilon_y{}^2 & 0 & 0 & 0 \\ 0 & 0 & \varepsilon_{\dot{x}}{}^2 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon_{\dot{y}}{}^2 & 0 \\ 0 & 0 & 0 & 0 & \varepsilon_n{}^2 \end{pmatrix}.$$

The entries of Q reflect the acceleration capabilities of a typical human. The noise for the number of people in the group, controlled by ε_n, accounts for people joining

or leaving the group without being noticed. The actual noise values used in our experiments are given in Sect. 4.6.

As mentioned before, we assume instantaneous rigidity for the shape of a group. Since the points in $\mathcal{P}$ are relative to the moving centroid, the point set remains unchanged, and $\mathcal{P}(k+1|k) = \mathcal{P}(k|k)$.

If two observations can be associated with a group track G, i.e., they both fall into the validation gate of G, the tracker can consider to *split* the track into two new tracks according to an interaction model (see Sect. 4.4). Since the actual partitioning in the split is unknown at this stage, two new predicted group tracks G_1 and G_2 are created by duplicating the predicted state and covariance of G. The same applies for the point set $\mathcal{P}$ and the label set $\mathcal{L}$. To make the label sets unique, we attach different indices to the label, e.g., a group with label set $\{0\}$ would split up into two groups with label sets $\{0-0\}$ and $\{0-1\}$. Again, the component of the state that represents the number of people in the group, n, is treated differently: the sum of people in the resulting groups must be equal to the original number of people. However, the actual partitioning is not known in the prediction step. Therefore, we use $n_1 = n_2 = n/2$, and reinitialize the state covariances of the new split tracks with C_0.

If the tracker considers to *merge* two group tracks G_i and G_j according to a track interaction model, the track prediction has to be computed accordingly. The predicted set of contour points of the merged group is the union of the two former point sets, $\mathcal{P}_{ij} = \mathcal{P}_i \cup \mathcal{P}_j$. The track states $\mathbf{x}_i$ and $\mathbf{x}_j$ of the merging group track represent the position and velocity of the centroids of the groups. Thus, the state of the merged track, $\mathbf{x}_{ij}$, is computed as the weighted mean of the original track states, using the number of points in the merging sets $\mathcal{P}_i$ and $\mathcal{P}_j$ as weights. The tracks before the merge are assumed to be independent. According to the summation and scaling laws for covariances, the covariance matrix of the merging track is the weighted mean of the original covariances with squared weights,

$$\mathbf{x}_{ij} = w_i \cdot \mathbf{x}_i + w_j \cdot \mathbf{x}_j \tag{17}$$

$$C_{ij} = {w_i}^2 \cdot C_i + {w_j}^2 \cdot C_j\,, \tag{18}$$

where $w_i = |\mathcal{P}_i|/|\mathcal{P}_{ij}|$ and $w_j = |\mathcal{P}_j|/|\mathcal{P}_{ij}|$. Note that this applies only for the first four components of $\mathbf{x}_{ij}$ and the upper-left 4×4 block of C_{ij}. The fifth component, namely the group size n_{ij}, is excluded, since the number of people in the merging groups naturally add up to $n_{ij} := n_i + n_j$. Consequently, the corresponding uncertainty values are summed up as well. Finally, the label set of the new group is the union of the label sets of the original tracks, $\mathcal{L}_{ij} = \mathcal{L}_i \cup \mathcal{L}_j$. To remove redundant labels, an optional pruning can be done in this step: whenever all tracks that resulted from a split have merged again, the additional indices added in the split step can be removed, e.g., when the groups with labels $\{0-0\}$ and $\{0-1\}$ merge, they can be labeled $\{0\}$ again. Although this can remove split and merge events from the history represented by the labeling, it keeps the semantic information consistent.

4.2.4 Group-to-Observation Assignment Probability

For data association we need to calculate the probability that an observed cluster $\mathcal{Z}_i$ belongs to a predicted group $G_j = \langle \mathbf{x}_j(k+1|k), C_j(k+1|k), \mathcal{P}_j \rangle$. Therefore, we are looking for a distance function $d(\mathcal{Z}_i, G_j)$ that, unlike the Mahalanobis distance used by Mucientes *et al.* [26], accounts for the shape of the observation cluster $\mathcal{Z}_i$ and the contour $\mathcal{P}_j$ of the group, rather than just for their centroids. To this end, we use a variant of the Hausdorff distance. As the regular Hausdorff distance is the *longest* distance between points on two contours, it tends to be too sensitive to large variations in depth that can occur in range data. This motivates the use of the minimum average Hausdorff distance [39] that computes the minimum of the averaged distances between contour points as

$$d_{\mathrm{HD}}(\mathcal{Z}_i, G_j) = \min\left\{d(\mathcal{Z}_i, \mathcal{P}_j), d(\mathcal{P}_j, \mathcal{Z}_i)\right\}, \tag{19}$$

where $d(\mathcal{Z}_i, \mathcal{P}_j)$ is the directed average Hausdorff distance from $\mathcal{Z}_i$ to $\mathcal{P}_j$,

$$d(\mathcal{Z}_i, \mathcal{P}_j) = \frac{1}{|\mathcal{Z}_i|} \sum_{\mathbf{z}_l \in \mathcal{Z}_i} \min_{\mathbf{x}_{\mathcal{P}_j} \in \mathcal{P}_j} \left\{D(\nu_{lj}, S_{lj})\right\}. \tag{20}$$

Since we deal with uncertain entities, we calculate the distance $d(\mathcal{Z}_i, \mathcal{P}_j)$ using the Mahalanobis distance

$$D(\nu_{lj}, S_{lj}) = \sqrt{{\nu_{lj}}^T S_{lj}^{-1} \nu_{lj}}, \tag{21}$$

with ν_{lj} being the innovation and S_{lj} being the innovation covariance between a point $\mathbf{z}_l \in \mathcal{Z}_i$ and contour point $\mathbf{x}_{\mathcal{P}_j}$ of the predicted set $\mathcal{P}_j$ transformed into the sensor frame. More precisely, these two terms are given as

$$\nu_{lj} = \mathbf{z}_l - (H\mathbf{x}_j(k+1|k) + \mathbf{x}_{\mathcal{P}_j}) \tag{22}$$

$$S_{lj} = HC_j(k+1|k)H^T + R_l, \tag{23}$$

where $H = \left(\begin{smallmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{smallmatrix}\right)$ is the measurement Jacobian and R_l the 2×2 observation covariance whose entries reflect the noise in the measurement process of the range finder.

The probability that cluster $\mathcal{Z}_i$ originates from group track G_j is finally given by a zero-centered Gaussian,

$$\mathcal{N}_i = \frac{1}{2\pi\sqrt{\det(S_{lj})}} \exp\left(-\tfrac{1}{2} d^2_{\mathrm{HD}}(\mathcal{Z}_i, G_j)\right). \tag{24}$$

4.2.5 Group-to-Group Assignment Probability

To determine the probability that two groups G_i and G_j merge, we compute the distance between their closest contour points in a Mahalanobis sense. In doing so, we have to account for the clustering distance d_P, since we consider G_i and G_j to be one group as soon as their contours come closer than d_P. Let $\Delta\mathbf{x}_{\mathcal{P}_{ij}} = \mathbf{x}_{\mathcal{P}_i} - \mathbf{x}_{\mathcal{P}_j}$ be the

vector difference of two contour points of G_i and G_j, respectively. We then subtract d_P from $\Delta\mathbf{x}_{\mathcal{P}_{ij}}$ unless $\Delta\mathbf{x}_{\mathcal{P}_{ij}} \leq d_P$ for which $\Delta\mathbf{x}_{\mathcal{P}_{ij}} = \mathbf{0}$. Concretely, the modified difference becomes $\Delta\mathbf{x}'_{\mathcal{P}_{ij}} = \max(\mathbf{0},\, \Delta\mathbf{x}_{\mathcal{P}_{ij}} - d_P\,\mathbf{u}_{\mathcal{P}_{ij}})$ where $\mathbf{u}_{\mathcal{P}_{ij}} = \Delta\mathbf{x}_{\mathcal{P}_{ij}}/|\Delta\mathbf{x}_{\mathcal{P}_{ij}}|$.

To obtain a similarity measure that accounts for nearness of group contours *and* similar velocity, we augment $\Delta\mathbf{x}'_{\mathcal{P}_{ij}}$ by the difference in the velocity components,

$$\Delta\mathbf{x}^*_{\mathcal{P}_{ij}} = (\Delta\mathbf{x}'^T_{\mathcal{P}_{ij}},\, \dot{x}_i - \dot{x}_j\,,\, \dot{y}_i - \dot{y}_j)^T\,. \tag{25}$$

We now determine the statistical compatibility of two groups G_i and G_j according to the four-dimensional minimum Mahalanobis distance

$$d^2_{\min}(G_i, G_j) = \min_{\substack{\mathbf{x}_{\mathcal{P}_i}\in\mathcal{P}_i \\ \mathbf{x}_{\mathcal{P}_j}\in\mathcal{P}_j}} \left\{ D^2(\Delta\mathbf{x}^*_{\mathcal{P}_{ij}}, C_i{+}C_j) \right\}. \tag{26}$$

The probability that two groups actually belong together, is finally given by

$$\mathcal{N}_{ij} = \frac{1}{(2\pi)^2\sqrt{\det(C_i{+}C_j)}} \exp\left(-\tfrac{1}{2}d^2_{\min}(G_i, G_j)\right). \tag{27}$$

In this formulation, only the upper-left 4×4 blocks of C_i and C_j are used, which excludes the group size estimate and the corresponding uncertainties from data association. In future work, these could be included as well.

4.3 Tracking Cycle

This section describes the steps in the cycle of our Kalman filter-based group tracker. An overview is given by the flow diagram in Fig. 13. The structure differs from a regular tracker in the additional steps *model generation*, *track re-prediction*, and *re-clustering*.

- *State prediction:* In this step, the states of all existing group tracks are predicted under the assumption that they continue without interacting with other tracks, i.e., without splits or merges. See Sect. 4.2.3 for details.
- *Observation:* As described in Sect. 4.1, this step involves grouping the laser range data into clusters $\mathcal{Z}$.
- *Model Generation:* Models are generated based on the predicted group tracks and the clusters $\mathcal{Z}$, see Sect. 4.4.
- *Re-prediction:* Based on the model hypotheses that postulate a split, merge, or continuation event for each track, groups are re-predicted using these hypotheses so as to reflect the respective model, as explained in Sect. 4.2.3.
- *Re-clustering:* Re-clustering an observed cluster $\mathcal{Z}_i$ is necessary when it might have been produced by more than one group track, that is, it is in the gate of more than one track. If the model hypothesis postulates a merge for the involved tracks, nothing needs to be done. Otherwise, $\mathcal{Z}_i$ needs to be re-clustered, which

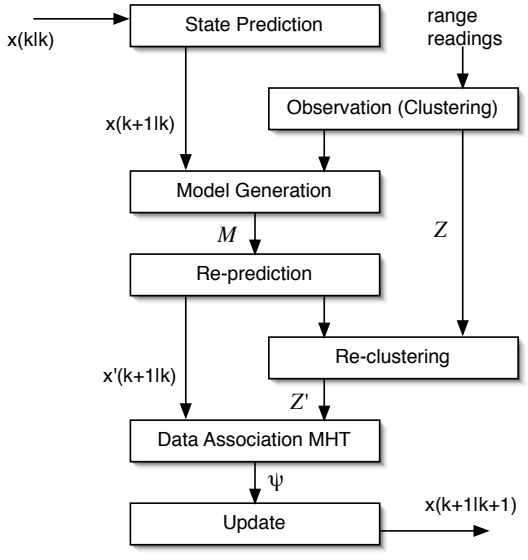

Fig. 13 Flow diagram of the tracking system. It differs from a regular tracker in the additional steps *model generation*, *re-prediction* and *re-clustering* (see explanations in section 4.3).

is done using a nearest-neighbor rule: those points $\mathbf{z}_l \in \mathcal{Z}_i$ that share the same nearest neighbor track are combined to a new cluster. This step follows from the uniqueness assumption, which is common in target tracking and according to which an observation can only be produced by a single target.

- *Data Association MHT:* This step involves the generation, probability calculation, and pruning of data association hypotheses that assign re-predicted group tracks to re-clustered observations. See Sect. 4.5.
- *Update:* Each group track G_j that has been assigned to a cluster $\mathcal{Z}_i$ is updated with a standard linear Kalman filter. We use an observation vector $\tilde{\mathbf{z}}_i = (\bar{\mathbf{z}}_i, n_{\text{hs}}(\mathcal{Z}_i))^T$, that contains both the centroid position $\bar{\mathbf{z}}_i$ of $\mathcal{Z}_i$ and the number of human-sized blobs $n_{\text{hs}}(\mathcal{Z}_i)$ in the cluster. The update is then given by

$$\mathbf{x}(k+1|k+1) = \mathbf{x}(k+1|k) + K\left(\tilde{\mathbf{z}}_i - \tilde{H}\mathbf{x}(k+1|k)\right) \tag{28}$$

$$C(k+1|k+1) = C(k+1|k) - K\tilde{H}C(k+1|k) \tag{29}$$

with K being the Kalman gain matrix and $\tilde{H}$ the corresponding measurement Jacobian,

$$K = C(k+1|k) \cdot \tilde{H}^T \left(\tilde{H}C(k+1|k)\tilde{H}^T + R_l\right)^{-1} \tag{30}$$

$$\tilde{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{31}$$

The contour points in $\mathcal{P}_j$ are replaced by the points in $\mathcal{Z}_i$ after being transformed into the reference frame of the posterior state $\mathbf{x}(k+1|k+1)$, as described in Sect. 4.2.2. Thereby, $\mathcal{P}_j$ always contains the most recent approximation of the group.

4.4 Model Generation and Model Probability

A model defines which group tracks are present at a certain time step, and thus assumes a particular state of the group formation process. New models, whose generation is described in this section, hypothesize about the evolution of that state. As this happens recursively, that is, based on the previous model of the last time index, the problem can thus be seen as a recursive clustering problem.

The space of possible model transitions is large since each group track can split into an unknown number of new tracks, or merge with an unknown number of other tracks. We therefore impose the gating condition for observations and tracks using the minimum average Hausdorff distance, thereby implementing a data-driven aspect into the model generation step:

- Multiple group tracks G_i can merge into one track only if there is an observation which is statistically compatible with all G_i.
- A group track can only split into multiple tracks that are all matched with observations in that very time step. Splits into occluded or obsolete tracks are not allowed.

Gating and statistical compatibility are both determined on a significance level α.

We further bound the possible number of model transitions as we assume that merge and split are binary operators. More precisely, we assume:

- At most two group tracks G_i, G_j can merge into one track at the same time.
- A track G_i can split at most into two tracks in one frame.
- A group track can not be involved in a split and a merge action at the same time.

These limitations are justified by the assumption that we observe the world much faster than the rate with which it evolves. This fact alleviates the impact of violations of the above assumptions: even if, for instance, a group splits into three subgroups at once, the tracker requires only two cycles to reflect this change.

A new model now defines for each group track if it is continued, split, or merged with another group track. The probability of a model is calculated using the constant prior probabilities for continuations and splits, p_C and p_S respectively, and the probability for a merge between two tracks G_i and G_j as $p_G \cdot \mathcal{N}_{ij}$. The latter term consists of a constant prior probability p_G and the group-to-group assignment probability $\mathcal{N}_{ij}$ defined in Sect. 4.2.5. Let N_C and N_S be the number of continued tracks and the number of split tracks in model M respectively, then the probability of M conditioned on the parent hypothesis Ω^{k-1} is

$$P(M|\Omega^{k-1}) = p_C^{N_C} \cdot p_S^{N_S} \prod_{G_i,G_j \in \Omega^{k-1}} (p_G \cdot \mathcal{N}_{ij})^{\delta_{ij}} \tag{32}$$

with δ_{ij} being 1 if G_i, G_j merge and 0 otherwise.

4.5 Multi-model

In this section we describe our adaptions and extensions of the original MHT by Reid [3] to a multi-model tracking approach that hypothesizes over both, data associations and models (as defined in the previous sections).

Let Ω_i^k be the i-th hypothesis at time k and $\Omega_{p(i)}^{k-1}$ its parent. Let further $\psi_i(k)$ denote a set of assignments that associate predicted tracks in $\Omega_{p(i)}^{k-1}$ to observations in $Z(k)$. As there are many possible assignment sets given $\Omega_{p(i)}^{k-1}$ and $Z(k)$, there are many children that can branch off a parent hypothesis, each with a different $\psi(k)$. This makes up an exponentially growing hypothesis tree.

The multi-model MHT introduces an intermediate tree level for each time step, on which models spring off from parent hypotheses (Fig. 14). In each model branch, the tracks of the parent hypothesis are first re-predicted to implement that particular model and then assigned to the (re-clustered) observations. Possible assignments for observations are existing tracks that *match* with existing tracks, *false alarms* or *new tracks*. Using the generalized formulation described in Section 3 to deal with more than two track interpretation labels, tracks are interpreted as *matched*, *obsolete*, or *occluded*.

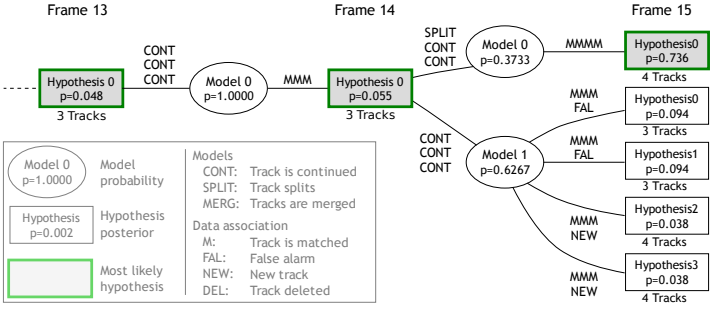

Fig. 14 The multi-model MHT. For each parent hypothesis, model hypotheses (ellipses) branch out and create their own assignment problems. In our application, models define which tracks of the parent hypothesis are continued, split, or merged. The tree shows frames 13 to 15 of figure 16. The split of group 1 between frames 14 and 15 is the most probable hypothesis after data association following model branch 0, although the continuation following model branch 1 is more probable (see the legend for details).

4.5.1 Assignment Set and Hypothesis Probability

The probability of a hypothesis in the multi-model MHT is calculated as follows. We compute the probability of a child hypothesis Ω_i^k given the observations from all time steps up to k, denoted by Z^k. According to the Markov assumption, it is the joint probability of the assignment set $\psi_i(k)$, the model M, and the parent hypothesis $\Omega_{p(i)}^{k-1}$, conditioned on the current observation $Z(k)$. Using Bayes rule, this can be expressed as the product of the data likelihood with the joint probability of assignment set, model and parent hypothesis

$$\begin{aligned} &P(\Omega_i^k|Z^k) \\ &= P(\psi, M, \Omega_{p(i)}^{k-1}|Z(k)) \qquad (33) \\ &= \eta \cdot P(Z(k)|\psi, M, \Omega_{p(i)}^{k-1}) \cdot P(\psi, M, \Omega_{p(i)}^{k-1}). \qquad (34) \end{aligned}$$

By using conditional probabilities, the third term on the right hand side can be factorized into the probabilities of the assignment set, the model, and the parent hypothesis

$$P(\psi, M, \Omega_{p(i)}^{k-1}) = P(\psi|M, \Omega_{p(i)}^{k-1}) \cdot P(M|\Omega_{p(i)}^{k-1}) \cdot P(\Omega_{p(i)}^{k-1}). \qquad (35)$$

The third factor in this product is known from the previous iteration, whereas the second factor represents the model probability derived in Sect. 4.4.

It remains to specify the first factor which is the probability of the assignment set ψ. The set ψ contains the assignments of observed clusters $\mathcal{Z}_i$ and group tracks G_j either to each other or to one of their possible labels listed above. Assuming independence between observations and tracks, the probability of the assignment set is the product of the individual assignment probabilities. Namely, they are p_M for matched tracks, p_F for false alarms, p_N for new tracks, p_O for tracks found to be occluded, and p_T for obsolete tracks scheduled for termination. If the number of new tracks and false alarms follow a Poisson distribution (as assumed by Reid [3]), the probabilities p_F and p_N have a sound physical interpretation as $p_F = \lambda_F V$ and $p_N = \lambda_N V$, where λ_F and λ_N are the average rates of events per volume multiplied by the observation volume V (the field of view of the sensor). The probability for an assignment ψ given a model M and a parent hypothesis Ω^{k-1} is then computed as

$$P(\psi|M, \Omega^{k-1}) = p_M^{N_M}\ p_O^{N_O}\ p_T^{N_T}\ \lambda_F^{N_F} \lambda_N^{N_N}\ V^{N_F+N_N}, \qquad (36)$$

where the Ns are the number of assignments to the respective labels in ψ.

Thanks to the independence assumption, also the data likelihood $P(Z(k)|\psi, M, \Omega_{p(i)}^{k-1})$ is computed by the product of the individual likelihoods of each observation cluster $\mathcal{Z}_i$ in $Z(k)$. If ψ assigns an observation $\mathcal{Z}_i$ to an existing track, we assume the likelihood of $\mathcal{Z}_i$ to follow a normal distribution, given by Eq. 24. Observations that are interpreted as false alarms and new tracks are assumed to be uniformly distributed over the observation volume V, yielding a likelihood of $1/V$. The data likelihood then becomes

$$P(Z(k)|\psi, M, \Omega^{k-1}) = \left(\tfrac{1}{V}\right)^{N_N+N_F} \prod_{i=1}^{N_Z} \mathcal{N}_i^{\delta_i}, \tag{37}$$

where δ_i is 1 if $\mathcal{Z}_i$ has been assigned to an existing track, and 0 otherwise.

Substitution of Eqs. (32), (36), and (37) into Eq. (33) leads, like in the original MHT approach, to a compact expression, independent on the observation volume V.

Finally, normalization is performed yielding a true probability distribution over the child hypotheses of the current time step. This distribution is used to determine the current best hypothesis and to guide the pruning strategies.

4.5.2 Hypothesis Pruning

Pruning is essential in implementations of the MHT algorithm, as otherwise the number of hypotheses grows boundless. The following strategies are employed:

K-best branching: instead of creating all children of a parent hypothesis, the algorithm proposed by Murty [31] generates only the K most probable hypotheses in polynomial time. We use the multi-parent variant of Murty's algorithm, mentioned in [40], that generates the global K best hypotheses for all parents.

Ratio pruning: a lower limit on the ratio of the current and the best hypothesis is defined. Unlikely hypotheses with respect to the best one, being below this threshold, are deleted. Ratio pruning overrides K-best branching in the sense that if the lower limit is reached earlier, less than K hypotheses are generated.

N-scan back: the N-scan-back algorithm considers an ancestor hypothesis at time $k-N$ and looks ahead in time onto all children at the current time k (the leaf nodes). It only keeps the subtree at $k-N$ with the highest sum of leaf node probabilities. All other branches at $k-N$ are discarded.

More details on these pruning strategies can be found in the work of Cox and Hingorani [29].

4.6 Experimental Evaluation

To analyze the performance of our system, we collected two data sets in the entrance hall of a university building, shown in Fig. 15. We used a Pioneer II robot equipped with a SICK laser scanner mounted at 30 cm above floor, scanning at 10 fps. In two unscripted experiments (dataset 1 with a stationary robot and dataset 2 with a moving robot), up to 20 people are in the field of view of the sensor. They form a large variety of groups during social interaction, move around, stand together and jointly enter and leave the hall, see Fig. 16.

To obtain ground truth information, we labeled each single range reading. Beams that belong to a person receive a person-specific label, other beams are labeled as non-person. These labels are kept consistent over the entire duration of the data sets. People that socially interact with each other (derived by observation) are said to belong to a group with a group-specific label. Summed over all frames, the ground

Fig. 15 Space where we have recorded the datasets for our experiments.

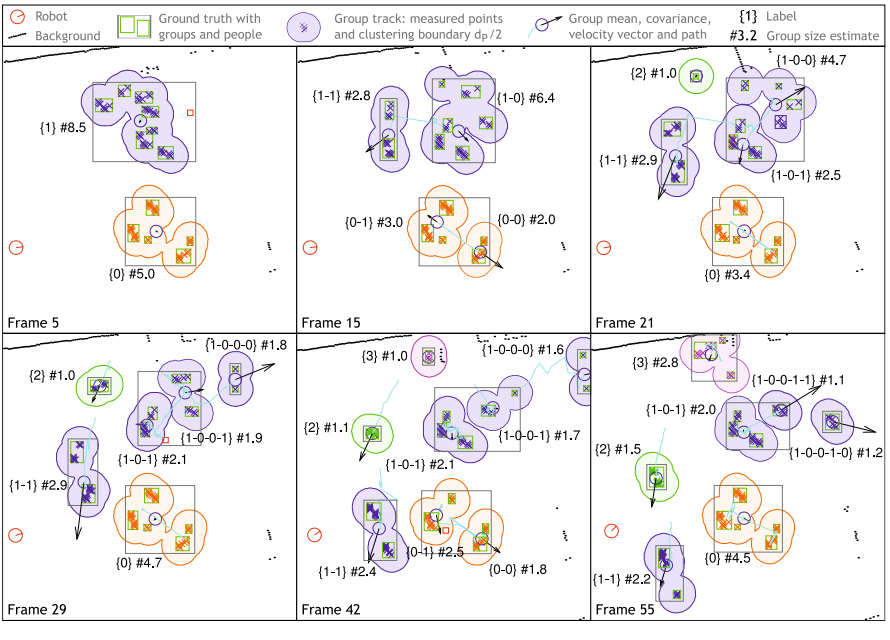

Fig. 16 Tracking results from the second data set. In frame 5, two groups are present. In frame 15, the tracker has correctly split group 1 into 1-0 and 1-1 (see Fig. 14). Between frames 15 and 29, group 1-0 has split up into groups 1-0-0 and 1-0-1 and split up again. New groups, labeled 2 and 3, enter the field of view in frames 21 and 42 respectively.

Table 11 Summary of the two datasets used in the experiments.

	Dataset 1	Dataset 2
Number of frames	578	991
Avg. / max people	6.25 / 13	8.99 / 20
Avg. / max groups	2.60 / 4	4.16 / 8
Number of splits / merges	5 / 10	48 / 44
Number of new tracks / deletions	19 / 15	34 / 39

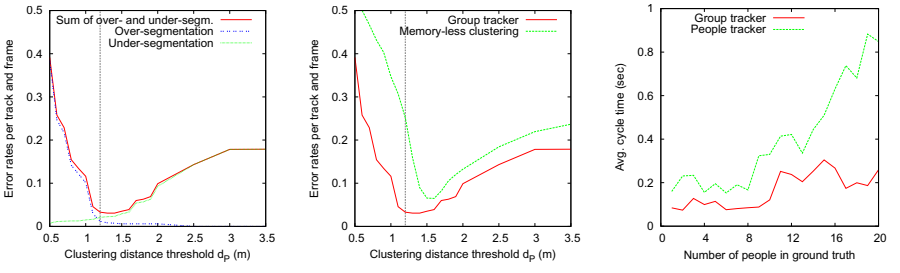

Fig. 17 *Left:* Clustering error of the group tracker as the sum of over-segmentation and under-segmentation error. The smallest error is achieved for a cluster distance of 1.3 m which is very close to the border of personal and social space according to the proxemics theory, marked at 1.2 m by the vertical line. *Middle:* Clustering error of the group tracker compared to memory-less single linkage clustering (without tracking). *Right:* Average cycle time for the group tracker versus a tracker for individual people plotted against the ground truth number of people.

truth contains 5,629 labeled groups and 12,524 labeled people.[1] For further details, see Tab. 11.

The ground truth data is used for performance evaluation and to learn the parameters of our tracker. The values, determined by counting the related events in the ground truth and dividing by the total number of these events, are $p_M = 0.79$, $p_O = 0.19$, $p_T = 0.02$, $p_F = 0.06$, $p_N = 0.02$ for the data association probabilities, and $p_C = 0.63$, $p_S = 0.16$, $p_G = 0.21$ for the group formation probabilities. When evaluating the performance of the tracker, we separated the data into a training set and a validation set to avoid overfitting.

The state uncertainty for new tracks is given by $\sigma_x = \sigma_y = 0.1$, $\sigma_{\dot{x}} = \sigma_{\dot{y}} = 0.5$, and $\sigma_n = 0.2$. The noise parameter for the motion model are given by $\varepsilon_x = \varepsilon_y = 0.2$, $\varepsilon_{\dot{x}} = \varepsilon_{\dot{y}} = 0.3$, and $\varepsilon_n = 0.1$.

Six frames of the current best hypothesis from the second dataset are shown in Fig. 16. The corresponding hypothesis tree for frame 15 is shown in Fig. 14. The sequence exemplifies movement and formation of several groups.

[1] Data sets, ground truth and result videos are available online at http://www.informatik.uni-freiburg.de/~lau/grouptracking

4.6.1 Clustering Error

This section analyzes how well the presented group tracker can recover the true group formation processes, i.e., which people actually belong together according to their social interaction as encoded in the ground truth.

We compute the clustering error of the tracker using the ground truth information on a per-beam basis. This is done by counting how often set of points $\mathcal{P}$ of a track contains too many or wrong points (under-segmentation) and how often $\mathcal{P}$ is missing points (over-segmentation). Two examples for over-segmentation errors can be seen in Fig. 16, where group 0 and group 1-0 are temporarily over-segmented, compared to the ground truth which is visualized with a rectangle. However, from the history of group splits and merges stored in the group labels, the correct group relations can be determined in such cases.

For the first dataset, the clustering error rates for under-segmentation, over-segmentation, and the sum of both are shown in Fig. 17 (left), plotted against the clustering distance d_P.

We compare the clustering performance of our group tracker with a memory-less group clustering approach, which performs single-linkage clustering of the range data as described in Sect. 4.1 without using a tracking framework. The result is shown in Fig. 17 (middle).

The minimum clustering error of 3.1% is achieved by the tracker at $d_P = 1.3\,m$. The minimum error for the memory-less clustering is 7.0%, more than twice as high. In the second dataset, the error rates are higher due to the larger number of occlusions and the increased complexity in group interactions. Here, the minimum clustering error of the tracker is 9.6% while the error of the memory-less clustering reaches 20.2%, again more than twice as high.

To further investigate situations where tracking results differ from memory-less clustering, we recorded laser data of groups of people walking and passing in a corridor. An example is shown in Fig. 18, where one person passes between a group of two people. The memory-less approach would merge them immediately while the tracking approach, accounting for the velocity information, correctly keeps the groups apart by using re-clustering. This result shows that the group tracking problem is a *recursive* clustering problem that requires integration of information over time.

In the light of the proxemics theory the result of a minimal clustering error at 1.3 m is noteworthy. The theory predicts that when people interact with friends, they maintain a range of distances between 45 to 120 cm called personal space. When engaged in interaction with strangers, this distance is larger. As our data contains students who tend to know each other well, the result appears consistent with the findings of Hall.

4.6.2 Tracking Efficiency

When tracking groups of people rather than individuals, the assignment problems in the data association stage are of course smaller. At the same time, the introduction of an additional tree level, on which different models hypothesize over different group

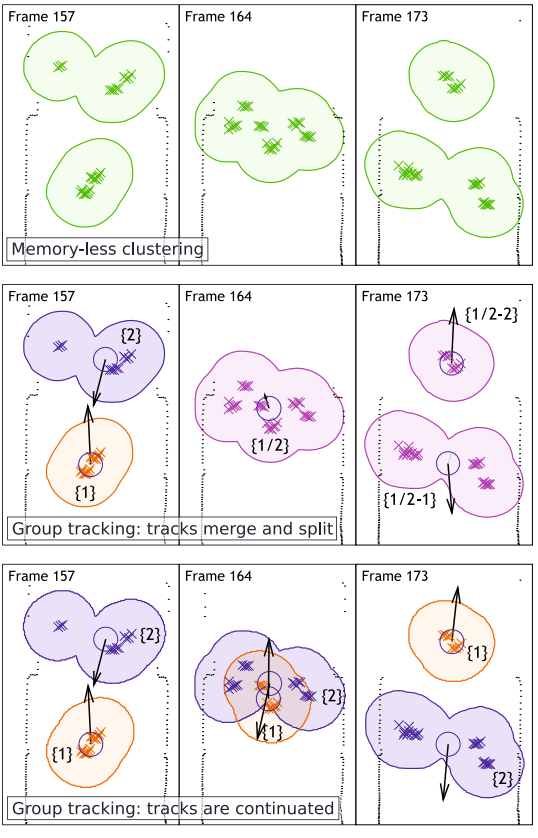

Fig. 18 One person crosses a group of two people. Since the groups interweave, memory-less clustering (top) unifies the two groups. Our group tracker can also create a model that postulates a merge of the groups, followed later by a split (middle). However, the model hypothesis leading to the most probable hypothesis in this situation continues both tracks and triggers re-clustering (see Sect. 4.3). This way, the crossing groups are tracked correctly (bottom). For a legend, see Fig. 16.

formation processes, comes with additional computational costs. We therefore compare our system with a person-only tracker realized by inhibiting all split and merge operations and reducing the cluster distance d_P to the value that yields the lowest error for clustering single people given the ground truth. For the second dataset, the resulting average cycle times versus the ground truth number of people is shown in Fig. 17 (right). The plots are averaged over different k from the range of 2 to 200 at a scan-back depth of $N = 30$.

With an increasing number of people, the cycle time for the people tracker grows much faster than the cycle time of the group tracker. Interestingly, even for small numbers of people the group tracker is faster than the people tracker. This is due to occasional over-segmentation of people into individual legs tracks. Also, as mutual

occlusion of people in densely populated environments occurs frequently, the people tracker has to maintain many more occluded tracks than the group tracker, as occlusion of entire groups is rare. Also, the additional complexity of multiple models in the group tracker virtually disappears when the tracks are isolated due to the data-driven model generation.

This result clearly shows that our group tracking approach is more efficient. With an average cycle time of around 100 ms for up to 10 people on a Pentium IV at 3.2 GHz, the algorithm runs in real-time even with a non-optimized implementation.

4.6.3 Group Size Estimation

To evaluate the accuracy of our group size estimation approach, we define the error as the absolute difference between the estimated number of people in a group and the true value according to the labeled ground truth. For counting the number of human-sized clusters in a group as described in Sect. 4.1, a clustering threshold $d_{\text{hs}} = 0.3\,m$ is used.

For the first dataset, we find that the average error in group size estimation is 0.23 people with a standard deviation of 0.30. In the more complex dataset 2, the average error is 0.33 people with a standard deviation of 0.49. If the estimated group sizes are rounded to integers, the tracker determined the correct value in 88.9% of all cases in dataset 1 and in 84.3% for dataset 2.

If only deviations of more than one person are considered an error, the system was correct in 99.5% of all cases in dataset 1 and 97.5% in dataset 2.

4.7 Summary

In this section, we presented a multi-model hypothesis tracking approach to track groups of people. We extended the original MHT approach to incorporate model hypotheses that describe track interaction events that go beyond what data association can express. In our application, models encode the formation of groups during split, merge, and continuation events. We further introduced a representation of groups that includes their shape, and employed the minimum average Hausdorff distance to account for the shape information when calculating association probabilities.

The proposed tracker has been implemented and tested using a mobile robot equipped with a laser range finder. The experiments with up to 20 people forming groups of different sizes demonstrated that the system is able to robustly track groups of people as they undergo complex formation processes. Given ground truth data reflecting true interactions of people with over 12,000 labeled occurrences of people and groups, the experiments showed that the tracker could reproduce such processes with a low clustering error and estimate the number of people in groups with high accuracy. They also showed that in comparison with a memoryless single-frame clustering, our system performs significantly better in determining which people form a group.

5 Conclusion

This article presented techniques for detecting and tracking people and groups of people with a mobile robot. The problem has been addressed using two-dimensional range data, as they are robust over the wide range of conditions that service robots encounter in real-world applications.

This sensor modality poses novel problems for people detection and tracking that have been addressed in this article. First, and in contrast to past approaches in the literature that mostly used hand-tuned classifiers, we presented a boosting approach based on AdaBoost to learn a strong classifier for people in range data. The algorithm combines a set of weak classifier computed from simple geometrical and statistical features for groups of neighboring laser beams that correspond to legs of people. The method has been implemented and applied in cluttered office environments. In practical experiments carried out in different environments we obtained worst-case detection rates of over 90%, clearly outperforming an alternative heuristic classifier.

Based on detected legs, we then addressed the tracking problem in the second section of the article. We employed a multi-target tracking framework that uses leg observations to infer the motion state of people. To achieve robust data association, we chose the Multiple-Hypothesis Tracking (MHT) framework and extended it to incorporate adaptive occlusion probabilities. The experiments using a stationary and a moving robot demonstrate that the approach was able to robustly track multiple people based on observations of their legs, also over lengthy occlusion events. The results further showed that our adaptive approach outperforms an MHT with fixed occlusion probabilities, since the latter overly delays track deletion and produces a high level of ambiguity coupled with an explosion of the number of hypotheses.

For the case when many people are present in an environment, we addressed the group tracking problem in the third section of this article. In this approach, the joint state of groups of people can be tracked, which reduces computational costs and can also reveil information about the social relation between people. We posed the group tracking problem as a recursive multi-hypothesis model selection problem in which we hypothesize over both, the partitioning of tracks into groups (models) and the association of observations to tracks (assignments). In our application, models encode the formation of groups during split, merge, and continuation events. The experiments with up to 20 people forming groups of different sizes demonstrated that the system is able to robustly track groups of people as they undergo complex formation processes. Given ground truth data reflecting true interactions of people, the experiments showed that the tracker could reproduce such processes with a low clustering error and estimate the number of people in groups with high accuracy. The experiments also demonstrated the ability of the approach to recover the actual social grouping of interacting people when compared to the ground truth. Interestingly, it was found that the clustering threshold for detection that produces the best tracking results appears consistent with the Proxemics theory from social psychology.

The work presented here employed 2D range data from a planar laser scanner. Such sensors are well established in many robotics applications due to their

accuracy, robustness and ease-of-use. Future work in people detection and tracking will account for the increasing availability of inexpensive 3D range and RGB-D sensors. They offer the advantage of much denser information on targets, some of them even provide built-in RGB data that allow for on-line learning of target-individual appearance models.

References

1. Arras, K.O., Mozos, Ó.M., Burgard, W.: Using boosted features for the detection of people in 2d range data. In: Proc. of the IEEE Int. Conference on Robotics and Automation, ICRA 2007, Rome, Italy (2007)
2. Arras, K.O., Grzonka, S., Luber, M., Burgard, W.: 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 2008), Pasadena, USA (2008)
3. Reid, D.B.: An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control AC-24(6), 843–854 (1979)
4. Hall, E.: Handbook of Proxemics Research. Society for the Anthropology of Visual Communications (1974)
5. Lau, B., Arras, K.O., Burgard, W.: Multi-model hypothesis group tracking and group size estimation. International Journal of Social Robotics 2(1) (March 2010)
6. Fod, A., Howard, A., Mataric, M.J.: Laser-based people tracking. In: Proceedings of the IEEE International Conference on Robotics & Automation, ICRA (2002)
7. Kleinhagenbrock, M., Lang, S., Fritsch, J., Lömker, F., Fink, G.A., Sagerer, G.: Person tracking with a mobile robot based on multi-modal anchoring. In: IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), Berlin, Germany (2002)
8. Scheutz, M., McRaven, J., Cserey, G.: Fast, reliable, adaptive, bimodal people tracking for indoor environments. In: IEEE/RSJ Int. Conference on Intelligent Robots and Systems, Sendai, Japan (2004)
9. Schulz, D., Burgard, W., Fox, D., Cremers, A.B.: People tracking with a mobile robot using sample-based joint probabilistic data association filters. International Journal of Robotics Research (IJRR) 22(2), 99–116 (2003)
10. Topp, E.A., Christensen, H.I.: Tracking for following and passing persons. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Alberta, Canada (2005)
11. Cui, J., Zha, H., Zhao, H., Shibasaki, R.: Tracking multiple people using laser and vision. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Alberta, Canada (2005)
12. Xavier, J., Pacheco, M., Castro, D., Ruano, A.: Fast line, arc/circle and leg detection from laser scan data in a player driver. In: Proc. of the IEEE Int. Conference on Robotics & Automation, ICRA 2005 (2005)
13. Hähnel, R., Burgard, W., Thrun, S.: Map building with mobile robots in dynamic environments. In: Proc. of the IEEE Int. Conference on Robotics and Automation, ICRA (2003)
14. Viola, P., Jones, M.J.: Robust real-time object detection. In: Proceedings of IEEE Workshop on Statistical and Theories of Computer Vision (2001)
15. Treptow, A., Zell, A.: Real-time object tracking for soccer-robots without color information. Robotics and Autonomous Systems 48(1), 41–48 (2004)

16. Mozos, O.M., Stachniss, C., Burgard, W.: Supervised learning of places from range data using AdaBoost. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), Barcelona, Spain, April 2005, pp. 1742–1747 (April 2005)
17. Rottmann, A., Martínez Mozos, O., Stachniss, C., Burgard, W.: Place classification of indoor environments with mobile robots using boosting. In: Proc. of the National Conference on Artificial Intelligence (AAAI), Pittsburgh, PA, USA, pp. 1306–1311 (2005)
18. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Mach. Learn. 37(3), 297–336 (1999)
19. Premebida, C., Nunes, U.: Segmentation and geometric primitives extraction from 2d laser range data for mobile robot applications. In: Robótica 2005 - Scientific Meeting of the 5th National Robotics Festival, Coimbra, Portugal (April 2005)
20. Aloupis, G.: On computing geometric estimators of location. Ph.D. dissertation, School of Computer Science, McGill University (2001)
21. Arras, K.O.: Feature-based robot navigation in known and unknown environments. Ph.D. dissertation, Swiss Federal Institute of Technology Lausanne (EPFL), These No. 2765 (2003)
22. Song, Z., Chen, Y., Ma, L., Chung, Y.C.: Some sensing and perception techniques for an omnidirectional ground vehicle with a laser scanner. In: Proceedings of the 2002 IEEE International Symposium on Intelligent Control (2005)
23. Kluge, B., Köhler, C., Prassler, E.: Fast and robust tracking of multiple moving objects with a laser range finder. In: Proceedings of the IEEE Int. Conf. on Robotics and Automation (2001)
24. Zajdel, W., Zivkovic, Z., Kröse, B.J.A.: Keeping track of humans: Have I seen this person before? In: IEEE International Conference on Robotics and Automation, Barcelona, Spain (2005)
25. Schulz, D.: A probabilistic exemplar approach to combine laser and vision for person tracking. In: Proc. Robotics: Science and Systems, Philadelphia, USA (August 2006)
26. Mucientes, M., Burgard, W.: Multiple hypothesis tracking of clusters of people. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 692–697 (October 2006)
27. Taylor, G., Kleeman, L.: A multiple hypothesis walking person tracker with switched dynamic model. In: Proc. of the Australasian Conference on Robotics and Automation, Canberra, Australia (2004)
28. Cui, J., Zha, H., Zhao, H., Shibasaki, R.: Laser-based interacting people tracking using multi-level observations. In: IEEE/RSJ Int. Conference on Intelligent Robots and Systems, Beijing, China (2006)
29. Cox, I.J., Hingorani, S.L.: An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(2), 138–150 (1996)
30. Bar-Shalom, Y., Li, X.-R.: Multitarget-Multisensor Tracking: Principles and Techniques. YBS Publishing, Storrs (1995)
31. Murty, K.G.: An algorithm for ranking all the assignments in order of increasing cost. Operations Research 16, 682–687 (1968)
32. Khan, Z., Balch, T., Dellaert, F.: MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12) (2006)
33. McKenna, S.J., Jabri, S., Duric, Z., Rosenfeld, A., Wechsler, H.: Tracking groups of people. Computer Vision and Image Understanding 80(1), 42–56 (2000)
34. Gennari, G., Hager, G.D.: Probabilistic data association medhods in visual tracking of groups. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2004)

35. Bose, B., Wang, X., Grimson, E.: Multi-class object tracking algorithm that handles fragmentation and grouping. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
36. Joo, S.-W., Chellappa, R.: A multiple-hypothesis approach for multiobject visual tracking. IEEE Transactions on Image Processing 16(11), 2849–2854 (2007)
37. Lau, B., Arras, K.O., Burgard, W.: Tracking groups of people with a multi-model hypothesis tracker. In: International Conference on Robotics and Automation (ICRA), Kobe, Japan (2009)
38. Hartigan, J.A.: Clustering Algorithms. John Wiley & Sons (1975)
39. Dubuisson, M.P., Jain, A.K.: A modified Hausdorff distance for object matching. In: Intl. Conference on Pattern Recognition, Jerusalem, Israel, vol. 1, pp. A:566–A:568 (1994)
40. Cox, I.J., Miller, M.L.: On finding ranked assignments with application to multi-target tracking and motion correspondence. IEEE Trans. on Aerospace and Electronic Systems 31(1), 486–489 (1995)