Using Boosted Features for the Detection of People in 2D Range Data

Kai O. Arras   Óscar Martínez Mozos   Wolfram Burgard

University of Freiburg, Department of Computer Science, D-79110 Freiburg

{`arras`|`omartine`|`burgard`}`@informatik.uni-freiburg.de`

Abstract— This paper addresses the problem of detecting people in two dimensional range scans. Previous approaches have mostly used pre-defined features for the detection and tracking of people. We propose an approach that utilizes a supervised learning technique to create a classifier that facilitates the detection of people. In particular, our approach applies AdaBoost to train a strong classifier from simple features of groups of neighboring beams corresponding to legs in range data. Experimental results carried out with laser range data illustrate the robustness of our approach even in cluttered office environments.

I. INTRODUCTION

Detecting people is a key capacity for robots that operate in populated environments. Knowledge about presence, position, and motion state of people will enable robots to better understand and anticipate intentions and actions.

In this paper, 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 as they provide a large field of view and, opposed to vision, are mainly independent from 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 this work 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, our approach uses 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 the detecting moving blobs that appear as local minima in the range image [1], [2], [3], [4]. 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 [5] extend the method of Schulz *et*
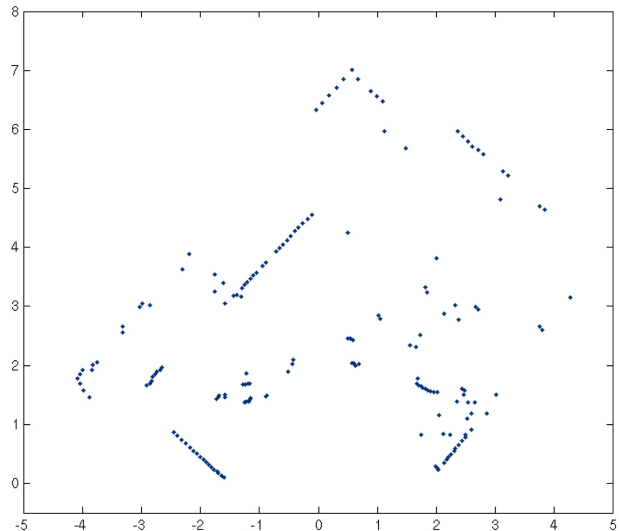

Fig. 1.   Where are the people? Example scan from a typical office.

al. [4] 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.* [6] 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.* [7]. 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.* [8] 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

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.

the entire groups according to their properties.

AdaBoost has been successfully used as a Boosting algorithm in different applications for object recognition. Viola and Jones [9] boost simple features based on grey level differences to create a fast face classifier using images. Treptow *et al.* [10] use the AdaBoost algorithm to track a ball without color information in the context of RoboCup. Further, Martínez Mozos *et al.* [11] 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.* [12] 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.

This paper is organized as follows. After briefly introducing boosting as a supervised approach for learning a classifier in the following section, we will discuss the features used by our approach in Section III. Finally, Section IV contains the experimental results obtained with the learned classifier.

II. 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 [13]. 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 selects repeatedly 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 increasing 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.

Throughout this work we use the approach presented by Viola and Jones [9] in which the weak classifiers depend on

TABLE I
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 $F(e) = \sum_{t=1}^{T} \alpha_t h_t(e)$.

single-valued features f_j and have the form

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

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 [11]. The final AdaBoost algorithm modified for the concrete task of this work is shown in Table I.

III. 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 [14] 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, in which there are different objects to be classified, one could learn decision lists as successfully applied by Martínez Mozos *et al.* [11] in the context of place labeling with mobile robots.

We define a feature f as a function $f : S \rightarrow \Re$ that takes a segment S as an argument and returns a real value. Here, 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 [15], 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 [16]. 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 $x = (x_c \quad y_c \quad x_c^2 + x_c^2 - r_c^2)^T$ 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.* [17]. 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 $\mathbf{x_A}\mathbf{x_B}\mathbf{x_C}$ 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.
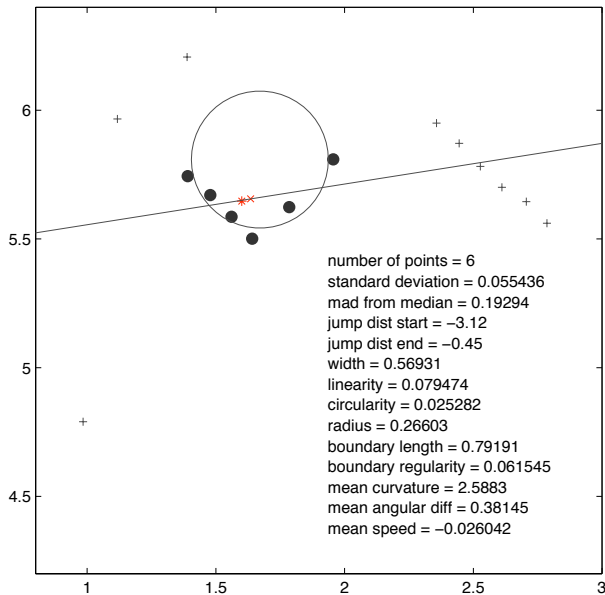
number of points = 6
standard deviation = 0.055436
mad from median = 0.19294
jump dist start = −3.12
jump dist end = −0.45
width = 0.56931
linearity = 0.079474
circularity = 0.025282
radius = 0.26603
boundary length = 0.79191
boundary regularity = 0.061545
mean curvature = 2.5883
mean angular diff = 0.38145
mean speed = −0.026042

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.

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.

Some papers [1], [6], [5] report the use of additional conditions on the distance between blobs, typically to associate two legs to the same person. We deliberately do not consider such conditions. The association of single legs to persons, especially when several people stand close together, is a complex data association problem. In our opinion, this problem can more robustly be solved by integrating multiple observations over time rather than directly on the level of the feature detector.

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

IV. EXPERIMENTS

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 III 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 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 III. 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.

A. 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 II. Only 1 from 240 segments (0.42%) corresponding

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

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

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

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

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 III.

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 IV shows the confusion matrix. Although the error rates slightly increase with respect to Tables II and III, 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.

B. 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 V 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.

C. 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 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 IV-A where segments from the corridor and office were used together as examples. The results of the classification are shown in Table VII. As the results indicate, our approach yields much better results than the heuristic approach.

D. 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 VIII. As can be seen, there is only a marginal improvement over the classifier without the motion feature (Table IV). 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.

E. 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 VI 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.

V. Conclusions

This paper addressed the problem of detecting people in laser range data. In contrast to previous approaches, which mostly used manually designed features, our approach applies the AdaBoost algorithm to learn a robust classifier. This classifier is learned from simple features and identifies groups of 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 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 VI).

In future work we plan to investigate how typical state estimation tasks like people tracking and mapping in dynamic environments can be improved using our robust classifier.

References

[1] A. Fod, A. Howard, and M. Mataric, "Laser-based people tracking," in *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[2] M. Kleinhagenbrock, S. Lang, J. Fritsch, F. Lömker, G. Fink, and G. Sagerer, "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.

[3] M. Scheutz, J. McRaven, and G. Cserey, "Fast, reliable, adaptive, bimodal people tracking for indoor environments," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.

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

[5] E. Topp and H. Christensen, "Tracking for following and passing persons," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Alberta, Canada, 2005.

[6] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, "Tracking multiple people using laser and vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Alberta, Canada, 2005.

[7] J. Xavier, M. Pacheco, D. Castro, and A. Ruano, "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'05)*, 2005.

[8] R. Hähnel, D. and, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, 2003.

[9] P. Viola and M. Jones, "Robust real-time object detection," in *Proc. of IEEE Workshop on Statistical and Theories of Computer Vision*, 2001.

[10] A. Treptow and A. Zell, "Real-time object tracking for soccer-robots without color information," *Robotics and Autonomous Systems*, vol. 48, no. 1, pp. 41–48, 2004.

[11] O. Martínez Mozos, C. Stachniss, and W. Burgard, "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.

[12] A. Rottmann, O. Martínez Mozos, C. Stachniss, and W. Burgard, "Place classification of indoor environments with mobile robots using boosting," in *Proc. of the National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, PA, USA, 2005, pp. 1306–1311.

[13] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.

[14] C. Premebida and U. Nunes, "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.

[15] G. Aloupis, "On computing geometric estimators of location," Ph.D. dissertation, School of Computer Science, McGill University, 2001.

[16] K. O. Arras, "Feature-based robot navigation in known and unknown environments," Ph.D. dissertation, Swiss Federal Institute of Technology Lausanne (EPFL), These No. 2765, 2003.

[17] Z. Song, Y. Chen, L. Ma, and Y. C. Chung, "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.