Deep Person Detection in 2D Range Data

Lucas Beyer[1], Alexander Hermans[1], Timm Linder[2], Kai O. Arras[3] and Bastian Leibe[1]

Abstract— Detecting humans is a key skill for mobile robots and intelligent vehicles in a large variety of applications. While the problem is well studied for certain sensory modalities such as image data, few works exist that address this detection task using 2D range data. However, a widespread sensory setup for many mobile robots in service and domestic applications contains a horizontally mounted 2D laser scanner. Detecting people from 2D range data is challenging due to the speed and dynamics of human leg motion and the high levels of occlusion and self-occlusion particularly in crowds of people. While previous approaches mostly relied on handcrafted features, we recently developed the deep learning based wheelchair and walker detector DROW. In this paper, we show the generalization to people, including small modifications that significantly boost DROW's performance. Additionally, by providing a small, fully online temporal window in our network, we further boost our score. We extend the DROW dataset with person annotations, making this the largest dataset of person annotations in 2D range data, recorded during several days in a real-world environment with high diversity. Extensive experiments with three current baseline methods indicate it is a challenging dataset, on which our improved DROW detector beats the current state-of-the-art.

I. INTRODUCTION

Robots that share spaces with people require the ability to detect, track and analyze humans for a variety of reasons including safety, user interaction, or efficient navigation through crowds [1], [2]. Self-driving cars [3] or last-mile delivery robots, for example, need to recognize pedestrians and particularly also vulnerable road users such as elderly people, children or people in wheelchairs. This recognition task is well understood for image data [4], [5], sometimes even specifically geared toward robust person detection [6], and with systems for pedestrian detection being commercially available for automotive use-cases. For other sensory modalities such as RGB-D or 2D/3D range data, often needed in robotics, a growing number of detection methods and datasets exist.

Many interesting approaches have been developed for detection in volumetric data, such as 3D LiDAR, which can largely be grouped into two types. The first type uses ordered projections of the point cloud data [7], [8]. The second, more recent type proposes novel neural network architectures which can consume unordered point cloud data and reason on them, *e.g.* PointNet [9] and VoxelNet [10].

[1]Lucas Beyer, Alexander Hermans and Bastian Leibe are with the Visual Computing Institute, RWTH Aachen University. `last@vision.rwth-aachen.de`

[2]Timm Linder is with Robert Bosch GmbH, Corporate Research, Stuttgart, Germany and with the University of Freiburg, Germany. `first.last@de.bosch.com`

[3]Kai O. Arras is with Robert Bosch GmbH, Corporate Research.

Despite the advent of increasingly performant and affordable 3D/RGB-D sensors, 2D horizontally mounted laser scanners are still a common part of the sensory setup on a large number of mobile service robots in human populated spaces, making them an interesting sensor for person detection. They are also one of the few certifiable sensors to ensure human safety *e.g.* for transportation platforms in intralogistics or hospitals, have large fields of view, high accuracy, and high levels of robustness in wide ranges of conditions. In contrast, on-board cameras have several drawbacks in many robotic use-cases including limited field of view, ambiguous scale, and likely detection failures when subjects are close to the sensor, an important case for human-robot interaction.

Existing 2D range data based person detectors can roughly be grouped into two types. The first type consists of jump-distance based clustering of the measured laser points, computation of many hand-crafted features on these clusters, and use of a machine learning approach to classify the cluster as either person or background using said features [11], [12], [13]. The second type does the same thing, although for legs as opposed to persons, by tracking legs over time, and defining persons as pairs of legs which satisfy some hand-designed heuristics [14], [15].

The latter type typically performs better on benchmark datasets and toy deployments, but the "two leg assumption" comes with some severe inherent drawbacks: 1) It does not readily extend to other classes, *e.g.* wheelchairs. 2) It only works when the legs are visible, which is neither true in many real-world scenarios due to people wearing coats, dresses, hijabs, saris, *etc.*, nor in many professional scenarios due to work uniforms such as lab coats. This is a perfect example of why it is desirable to collect a diverse dataset and directly learn from data instead of hand-crafting large parts of a system. Learning from 2D range data is difficult as evidenced by the relatively poorer performance of the jump distance-based methods when compared to the leg tracking methods. One big difference is that the latter often make use of rich temporal information. While deep learning has made impressive progress, also for detection on range data, so far our DROW [16] detector is the only deep learning based detector for 2D range data. DROW obtained state-of-the-art detection results for walking aids and has performed well in multiple real-world deployments.

In this paper we extend DROW to person detection. Given the multi-class nature of DROW, this is as straight forward as adding an additional class. However, given that persons in 2D range data are usually covered by few points, person detection proved to be a challenging task for the DROW

detector too.

For every point in a laser scan the DROW detector predicts a class and possibly a vote to the closest object center. A final post-processing step then converts votes to actual predictions. We improve this post-processing step, which now allows us to predict confidences for each detection and improves the overall performance. Additionally, we slightly increase our convolutional neural network (CNN) size, further boosting detection scores.

Orthogonal to these extensions, we experiment with a fully online temporal integration of a few previous scans and show that this results in a significant performance boost for person detection scores. Unlike a tracker, we do not perform any data association, but rather show how DROW's input pre-processing can be extended to include temporal context to be used by the network. We show how several fusion strategies in the network compare and obtain state-of-the-art results using our extended DROW detector.

To facilitate all our experiments, we introduce a new dataset of people annotations for our original DROW dataset. We hand annotate persons for every frame which we previously annotated with wheelchair and walker center points. To the best of our knowledge, this is the largest dataset for person detection in 2D range data recorded in real-world scenarios with over 10 hours of laser scan recordings.

We thoroughly evaluate top performing existing detectors on this dataset, including retraining and tuning hyperparameters on our validation set. Since person detection is arguably a bigger field than walking aid detection in the robotics community, with the addition of person annotations, the DROW dataset is now by far more interesting and relevant. Our final results and the complete DROW dataset will be shared with the community to foster further research in this area[1].

To summarize, our key contributions are as follows:

1) We extend the DROW detector to persons and propose several improvements boosting overall performance.
2) A temporal fusion extension to integrate past frames into the detector, achieving state-of-the-art detection results for persons and their walking aids.
3) A new large-scale "in the wild" indoor 2D laser dataset for person detection.
4) A thorough evaluation of current state-of-the-art person detectors on our dataset.

II. THE DROW DATASET

We introduce a large new set of person annotations for the existing DROW dataset [16], which was recorded with a SICK S300 laser scanner at 12.5 Hz and an angular resolution of 1/2 degree at 37 cm above ground, mounted on a moving platform[2]. To facilitate temporal approaches, we used the same annotation strategy as before: We divide the dataset into batches of 100 frames, annotating every fourth of these batches, while annotating every fifth frame within a batch.

[1] https://arxiv.org/abs/1804.02463
[2] For a detailed description of the data and the recording setup, see [16].

TABLE I: DROW Dataset overview

	Train	Validation	Test	Total
Sequences	78	30	5	113
Scans	341 138	74 744	48 131	464 013
Annotated Scans	17 665	3 919	2 428	24 012
Wheelchairs	14 455	5 595	1 970	22 020
Walkers	2 047	219	581	2 847
Persons	23 906	3 102	1 976	28 984
Wheelchairs by distance				
Walkers by distance				
Person by distance				
All by distance				

This results in 5% of all frames being annotated, stretched over long time-frames, allowing for interesting temporal approaches (see Section IV). We annotated exactly those frames which were previously annotated with wheelchairs and walkers, annotating persons that were walking on their own, as well as those that were pushing a walking aid. This means that a person pushing a walker results in two annotations: the person and the walker, while a person sitting in a wheelchair is only labeled as a wheelchair. We also considered persons sitting on normal chairs as persons, since these have a clearly different pattern in the data.

Table I shows an overview of the resulting dataset. All numbers given in the table refer to individual annotations and not to instances, and the bar plots show their distribution over the distance. Each bar represents a 1 m slice in the distance (15 bars for up to 15 m).

It should be noted that persons were significantly harder to annotate. By scrolling through a few frames and looking at the RGB images of the camera, it was fairly easy to find walking aid patterns in the scans. However, to annotate persons it was often necessary to focus on a single blob and "track" the pattern throughout a full batch of 100 frames in order to reliably tell which laser points represent a person. This clearly suggests that single frame detection of persons is a difficult task.

Compared to the few publicly available 2D laser range datasets for person detection at leg height, ours is to our best knowledge the largest in number of raw and annotated scans, as well as covered duration and therefore much better suited for the investigation of deep-learning methods. While [15] introduced a dataset for person tracking in 2D laser at leg height, their indoor multi-person dataset is limited to three sequences of 12 minutes total duration, significantly smaller than ours. Furthermore, due to the static background in the positive training data, their dataset is less suited for deep learning approaches which will easily overfit.

The home and reha datasets of [13] were recorded with a similar sensor setup and in a similar scenario as ours. While the number of annotated scans is similar, our raw scan count is larger by an order of magnitude and covers

a significantly longer period of time. Our average person count per annotated frame is twice as high, suggesting that our dataset is more challenging.

The Freiburg Main Station and City Center datasets [17] are around an order of a magnitude smaller in terms of total number of scans, and contain 2 to 4 times less annotated scans. The sensor was mounted on a static platform with limited background variability at a height of $80\,\mathrm{cm}$ where individual legs are usually not visible.

Our dataset shows interesting interactions between persons and their walking aids with varying backgrounds and a lot of real-world clutter, making it an interesting and challenging dataset.

III. METHOD AND EXTENSIONS

In order to give an overview of our proposed extensions, we first summarize the relevant parts of the DROW method. We refer the interested reader to [16] for a more detailed treatment. We directly show and discuss resulting precision-recall curves in the following sections, more details of our experimental setup are provided in Section VI.

A. DROW Baseline

The DROW detector, given a laser scan s will first "cut out" a window of fixed real-world size around each individual point i: $w_i = \mathrm{cut}(s, i)$. The opening-angle used by *cut* to determine which laser points are part of such a window of width ℓ at distance s_i is computed as $2\tan^{-1}(\frac{\ell}{2s_i})$. The *cut* operation additionally involves clamping of fore- and background as well as re-sampling the data in the window to a fixed number of points. Second, each *cutout* w_i is then fed to a CNN which predicts 1) whether that cutout belongs to an object of interest and, if so, 2) a vote regarding the location of the object's center relative to the cutout's location. Finally, all votes are collected and cast into a grid centered around the sensor origin, which is then smoothed before finding the maxima and reporting these as detections. The precision-recall curves of our re-implementation are shown as dotted lines (⋯⋯) in Figure 1.

B. Voting

In [16], votes of all classes were cast into an "agnostic" voting grid based on the agnostic *object*-probability $p(O|w) = \sum_{c \in \mathcal{C}} p(c|w_i)$ surpassing the detection threshold, and additionally into a class-specific grid according to their class. After smoothing and non-maximum suppression (NMS), the cells which correspond to maxima in the agnostic grid generate detections at their center using the class whose class-specific grid has the largest value in the corresponding cell. This has several drawbacks; for one, it discretizes the space of generated detections since only grid-centers are reported, and it does not assign class-distributions to detections, which are useful for other components down the line.

Joint voting. We propose an improved scheme which solves both these problems. We still accumulate all the votes into an agnostic voting grid using $p(O|w)$ and perform smoothing
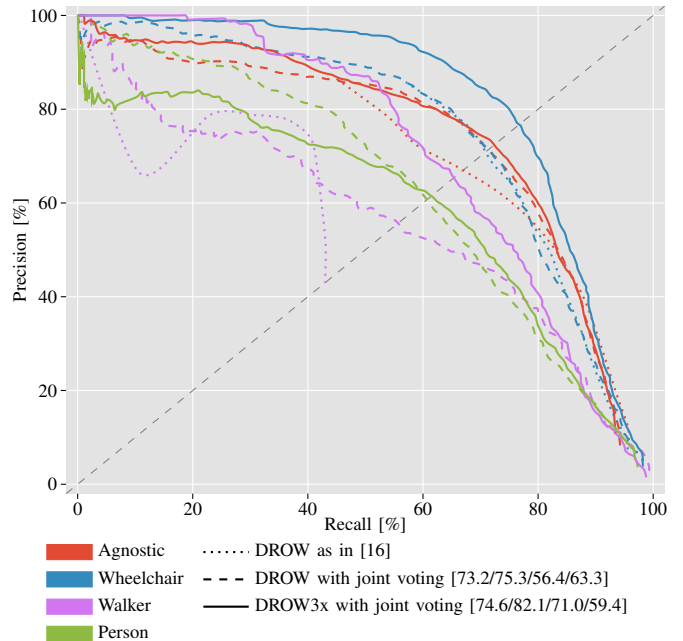

Fig. 1: Precision-recall curves showing the impact of our extensions. The numbers in brackets correspond to the AUC of the agnostic, wheelchair, walker, and person curves. These values are not computable for DROW, see discussion in Section III-B.

and NMS. However, we then assign each individual vote to the maximum in the agnostic grid which is closest to that vote, discarding spurious votes whose distance to the nearest maximum surpasses a certain radius. We then report a detection through the mean of its assigned votes, both for the location and the class distribution. This results in fully continuous detections as well as well-behaved class distributions. For evaluation purposes we simply report the probability of the dominant class, but in practice components processing the detections could make use of the full distribution. The precision-recall curves of this new voting scheme (now also including persons as a class) are represented by dashed lines (- - -) in Figure 1, showing a major improvement for walkers with wheelchairs being largely unaffected.

Split voting. We also experimented with an alternative voting mechanism where we did not use an agnostic voting grid, but only class-specific ones. The advantage here is that it is now possible to apply class-specific blurring. However, this then needs another NMS step on the reported detections in order to remove near exact duplicates of different classes. Split voting consistently outperformed joint voting by about 1%, however, it has multiple additional hyperparameters which severely impact performance. Since this method also is a little slower and we aim for a robust detector, we decided to use joint voting instead.

Knowing which points in the scan contributed to one detection has further advantages, one could for example estimate a confidence for the localization based on how the votes are distributed, or one could determine the spatial extent of the detected person. However, we leave these
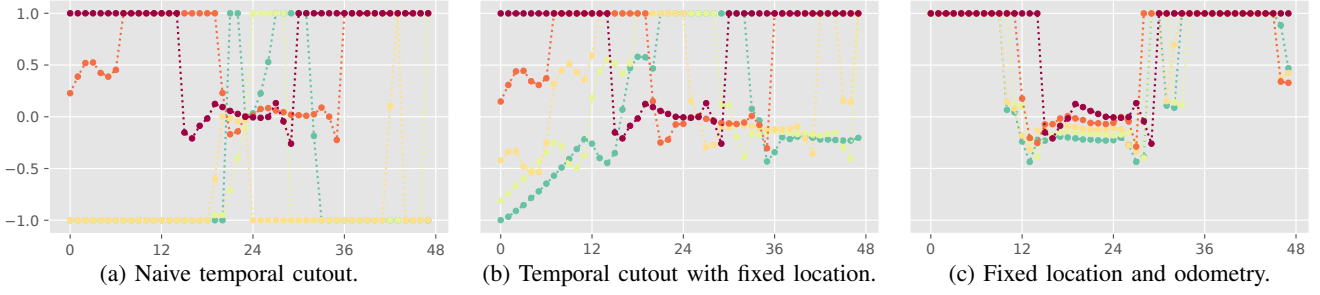
| | (a) Naive temporal cutout. | (b) Temporal cutout with fixed location. | (c) Fixed location and odometry. |

Fig. 2: Variants of temporal cutouts described in Section IV. All three cutouts correspond to the same point i in a scan at time t. All dots correspond to actual values of W_i^t, the color gradient moves from red at t into the past. This is the actual input seen by the CNN for that specific point i.

extensions to future work.

C. Network Architecture

We also update the network architecture using current best-practice, the main difference being that it is larger. This significantly improves the results, especially for the most difficult walker class, *c.f.* the solid lines (——) in Figure 1. The new network architecture, dubbed "DROW3x", consists of three stages of three zero-padded convolutional layers of filter size three each, initialized as in [18]. Batch-normalization [19] and Leaky-ReLU [20] (leak 0.1) nonlinearities are applied after all convolution operations. Each stage is followed by a max-pooling operation of size two and standard dropout [21]. A fourth, final stage consists of two more convolutional blocks as above followed by global average pooling, similarly to [22], [23], after which a single fully-connected layer computes the outputs, *i.e.* SoftMax probabilities and regressed votes. We have also experimented with various ResNet [23] and ResNeXt [24]-like architectures, but they did not lead to any further improvements while being significantly slower[3].

IV. TEMPORAL INTEGRATION

Another important cue for making sense of the low amount of information conveyed by a single laser scan is the change over time. The manual annotation, especially of persons, was only feasibly by scrolling through time and "tracking" each person individually in one's head. The way the DROW dataset was annotated allows for interesting ways of incorporating the time dimension into a method, although that was not taken advantage of in [16]. Here, we propose a few simple ways in which that temporal information can be used to great advantage, and we hope that these initial results will spur research into more elaborate ways of using time in detection. The general idea is to give the neural network a short snippet of the past in addition to the current measurement. Note that this is significantly different from and, in fact, orthogonal to tracking; the resulting method is still a detector. In order to do so in the context of the DROW detector, we need to decide on two things: 1) What is a "short

snippet of windows," and 2) where in the neural network is the time dimension fused?

Temporal cutouts. When generating the cutout w_i^t for a given point i of the scan at the current time t, one could use the cutouts for that same point at the previous times and collect these into a *temporal* cutout

$$W_i^t = \left[\text{cut}\left(s^t, i\right) \cdots \text{cut}\left(s^{t-T}, i\right) \right]. \quad (1)$$

This naive approach, however, is flawed: if the point i falls onto a leg at times t and $t-2$, but between the legs onto the background at time $t-1$, this approach would lead to the second cutout being centered on whatever is behind that person and thus giving arbitrary, wrong temporal context information as shown in Figure 2a. Instead, it is more meaningful to fix a cutout's location r according to the current time t, and use that fixed location for cutting out the same window at the previous times as temporal context:

$$W_i^t = \left[\text{cut}\left(s^t, i, r^t\right) \cdots \text{cut}\left(s^{t-T}, i, r^{t-T}\right) \right]. \quad (2)$$

This results in much more consistent windows, see also Figure 2b. However, more difficulties keeping the window fixed in real-world location arise when the robot turns: when turning, a point at index i in the scan at time t will be in another index j at time $t-1$. This difference $\Delta = i - j$ can be provided by most mobile platform's hardware through the odometry, and can be taken into account for the temporal cutouts as follows:

$$W_i^t = \left[\text{cut}\left(s^t, i, r^t\right) \cdots \text{cut}\left(s^{t-T}, i - \Delta^{t-T}, r^{t-T}\right) \right]. \quad (3)$$

The final, stabilized cutout is shown in Figure 2c. A further possible improvement would be to make use of the translational component of the odometry, however, translating a laser scan correctly is an ill-posed problem that we do not address in this work.

Temporal fusion. Now that we have a way to generate a meaningful temporal cutout W_i^t for each individual point i in the current scan s^t, the next question is how this temporal cutout is most effectively used by the neural network. The most obvious choice would be to simply stack the entries in the *channel* dimension of the network's input; this is also known as *early fusion* (EF). As has been shown most

---

[3]They did minimize the loss much better, but this did not translate to gains in precision-recall curves.
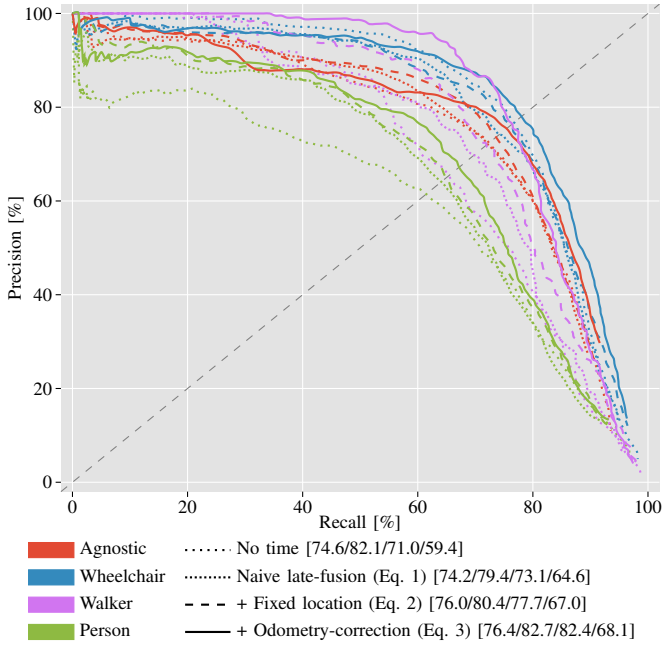
Fig. 3: Different ways of incorporating temporal context into DROW3x, as described in Section IV. The numbers in brackets correspond to the AUC of the agnostic, wheelchair, walker, and person curves.

prominently by [25], this is already a reasonable approach that allows the network to learn something about motion in the scene. Further research [26] has shown that fusing temporal information later into the network, *i.e.* *late fusion* (LF) can be more effective. We achieve this by replicating the first two blocks of the network in the time dimension, keeping the weights tied, and passing their sum as input to the third block of the network. This way, the network keeps the exact same amount of parameters, allowing for a direct comparison of the impact of time in Table II. It is likely that more extensive study of the fusion mechanism, including the use of recurrent neural networks (RNNs), will lead to even better results. This is not the major focus of our work and left for future work.

Figure 3 shows the progression of DROW3x with joint voting from not using time at all (⋯⋯) to the naive temporal cutouts (Eq. 1 ⋯⋯), the fixed-location cutouts (Eq. 2 - - -), finally leading to the fully "corrected" temporal cutouts (Eq. 3 ——). The main observation here is that while using time is *always* better than not using it, making the input to the neural network as consistent as possible helps even more.

V. BASELINES

We compare our proposed method against three state-of-the-art methods for person detection in 2D laser described in the following subsections. The segment-based person detector by Arras *et al.* [11], the ROS leg detector [14] and the joint leg tracker by Leigh *et al.* [15]. Figure 4 shows precision-recall curves for all baselines and our proposed approach (——) on the task of person detection. For each evaluated method, we also report percentage values for area

TABLE II: Comparing temporal fusion strategies.

	wheelchair			walker			person		
	AUC	p-F1	EER	AUC	p-F1	EER	AUC	p-F1	EER
No time	82.1	77.5	76.6	71.0	66.0	65.4	59.4	61.5	61.4
Naive Early-fusion	79.2	72.7	72.4	75.3	71.6	70.6	63.7	62.2	62.0
+ Fixed location	78.9	74.1	73.5	73.3	69.7	67.3	65.9	64.8	64.0
+ Odom correction	78.4	73.7	73.1	76.6	72.2	69.9	63.0	64.3	64.2
Naive Late-fusion	79.4	74.9	74.2	73.1	70.7	69.9	64.6	64.5	64.1
+ Fixed location	80.4	76.7	76.1	77.7	73.4	72.1	67.0	65.9	64.9
+ Odom correction	**82.7**	**78.7**	**78.1**	**82.4**	**78.9**	**76.4**	**68.1**	**68.1**	**67.2**

under the curve (AUC), equal-error rate (EER) and peak-F1 measure.

For this evaluation, we do not compare against the Gandalf detector [13], which was already part of the comparison of the original DROW paper [16]: it is a segment-classification based method, but uses distant-invariant features for the classification of segments which is similar to DROW's depth-robust preprocessing. While it is one of the few multi-class detectors for 2D range data, it is not straight forward to evaluate for person detection since the available code often produces leg detections, rather than person centroids, and requires an additional data association / tracking step to merge leg detections. Like the training code, this has not been made publicly available. Nevertheless, our DROW baseline outperformed the Gandalf detector in [16]. Finally, the resulting detections do not have confidences, nor is there a meaningful parameter to vary in order to create more than a single precision-recall point.

A. Segment-based person detector by Arras et al.

The original approach for person detection in 2D range data by Arras *et al.* [11], which forms the basis of the other two methods described below, combines a jump distance segmentation of the laser scan with a boosted classifier trained on hand-crafted geometric features. The jump distance segmentation threshold is configured such that two legs typically end up in the same segment.

We use our ROS-based reimplementation [27], in which the Adaboost classifier has been replaced by a better-performing random forest. We retrain the detector on the person annotations of the DROW training set, as the model from [27] assumed a different sensor mounting height of 70–80 cm where legs are often not visible. Automated hyperparameter tuning (Sec. VI) results in a segmentation threshold of 23 cm and a random forest with 128 trees of max depth 50. During training, a segment is regarded as foreground if over half of its points end up in a circle of 25 cm radius around the annotated person centroid, and as background if over half of the points in a segment do not fall into such a circle. All remaining segments are considered as ambiguous and not used for training. The resulting precision-recall curve is shown in Figure 4 (——).

As we observed during initial experiments that false positive person detections often occur at the left/right boundaries

of the sensor field of view, we incorporate an additional geometric feature that measures the angular difference of the segment center to the boundary of the sensor FOV, which improves performance in AUC by 2.6 percentage points (——).

B. ROS leg detector

This "detector" [14] (——) is often used for experiments in robotics as it is readily available as a standard ROS package with a pre-trained model. It builds on top of the geometric features from [11], but applies them to individual leg clusters using a random forest classifier. The detected legs are then associated and *tracked* over time using a Kalman filter-based tracker. For our evaluation, we treat the resulting tracks on a frame-by-frame level as if they were detections. Following the findings in [15], we reduce the leg reliability threshold from 0.7 to 0.3 as otherwise only very few person tracks are initiated.

C. Joint leg tracker by Leigh et al.

The joint leg tracker by Leigh *et al.* [15] is an improved and extended variant of [14] which jointly tracks two legs by allowing up to two associations of detected leg clusters with a person track. Similarly to the ROS leg detector, because our evaluation is based upon person centroids as opposed to individual legs, we treat the output of their tracker as detections by considering only non-occluded, matched tracks which have been associated with at least one leg detection in the current frame[4]. Detection confidences are computed from track confidence scores. We alter the original training method, which assumes that independent data recordings with positive and negative training examples are available, by assigning person annotations to potential leg clusters within a given radius, *e.g.* 25 cm. For the purpose of a fair evaluation, we disable their proposed online occupancy grid mapping extension[5], which is used to suppress false positive detections in walls and other static structures; however, *all* evaluated methods could similarly be extended to suppress false detections *e.g.* by using a static map of the environment as shown in [27].

In one experiment, *c.f.* Figure 4 (——), we use the provided model off-the-shelf with default hyperparameters, and see that it performs slightly better than the off-the-shelf ROS leg detector, but worse than the retrained Arras detector. In another experiment, we retrain the classifier and extensively tune 10 different hyperparameters, including leg association and track initiation/continuation thresholds, segmentation distance, confidence update factor as well as class weighting and person radius during training. This leads to significantly better results (——). The large difference in performance can partly be explained by the dependence of the learned model's features on the angular resolution of the sensor (originally 1/3 degree).

We can also see that the joint leg tracker, even after extensive automatic hyperparameter optimization, achieves a maximum recall of only around 75%, while both our proposed approach and the Arras detector can achieve up to 95% recall. Further experiments show that the method of Leigh outputs basically no person detections beyond a distance of around 7m, whereas both DROW3x and the detector by Arras still generate detections up to the maximum detection distance of 15m. The reason for this appears to be that at 7m distance and an angular resolution of 1/2 degree, a single leg cluster often consists of only two or less points, while some of the used geometric features by design require at least three points[6]. Our implementation of the Arras detector, which uses a similar set of geometric features, does not suffer from this issue because the entire person (including both legs) is considered as one segment upon which features are computed. We believe the limited maximum range is a significant limitation of the leg-tracking method of [15], as many robots are still equipped with 2D laser range finders with an angular resolution of 1/2 deg or less (*e.g.* [16], [13], [17], [12]), especially when considering the increasing adoption of low-cost, low-res 2D laser scanners in domestic applications.

VI. EXPERIMENTAL SETUP

For the experiments in this paper, we re-implemented our original DROW detector in PyTorch [28] from scratch. We now use the Adam [29] optimizer and fix a learning rate schedule starting at 10^{-3} for 40 epochs, then exponentially decaying to 10^{-6} within 10 epochs. Initial comparisons showed this not to affect the scores much, but it stabilizes results at the end of training and makes experiments comparable without the need to cherry-pick.

Unless stated otherwise, our DROW detector is always trained for three classes (persons, wheelchairs and walkers), however, to avoid cluttered plots we only show the precision-recall curves for persons in Figure 4. Initial experiments show that no significant improvements were obtained by training a person-only version of our detector.

We use Hyperopt [30] to tune most hyperparameters of all methods (except neural network training) on the validation set[7]. We optimize the sum of the AUCs of all available precision-recall curves. For our DROW baselines, the old voting mechanism does not produce dense precision-recall curves, thus making it difficult to compute AUCs. In this case, we optimize the sum of the peak-F1 scores.

During evaluation a detection is considered correct if the centroid lies within $0.5\,\mathrm{m}$ radius of an annotation. At most one detection can be assigned to each annotation.

VII. DISCUSSION

Looking at the results in Figure 4, one can distinguish between three types of methods.

---

[4]Preliminary experiments showed that inclusion of occluded tracks leads to AUC/EER scores that are worse by around 4 percentage points.

[5]We were also not able to run it reliably on all our sequences.

[6]We tried to set the values of affected features to a constant in such cases and retrain the classifier, but neither achieved better results, nor real-time performance (< 1 Hz) using a lower minimum point count per leg cluster.

[7]We allocate the same computational budget for each experiment.

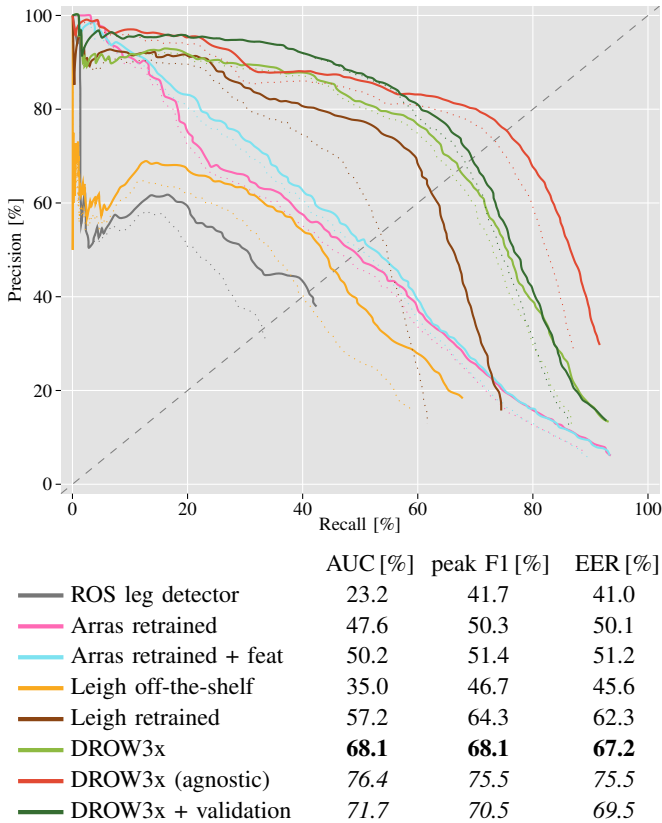|	AUC [%]	peak F1 [%]	EER [%]
ROS leg detector	23.2	41.7	41.0
Arras retrained	47.6	50.3	50.1
Arras retrained + feat	50.2	51.4	51.2
Leigh off-the-shelf	35.0	46.7	45.6
Leigh retrained	57.2	64.3	62.3
DROW3x	**68.1**	**68.1**	**67.2**
DROW3x (agnostic)	*76.4*	*75.5*	*75.5*
DROW3x + validation	*71.7*	*70.5*	*69.5*

Fig. 4: Different baselines and our proposed approach (DROW3x) evaluated on the person class of our test set. The thin dotted lines show the same experiment, using an evaluation radius of $0.3\,\text{m}$. Unlike all other curves the red line represents an agnostic evaluation, meaning we use detections and annotations of all classes jointly.

Leg-tracking based methods which detect individual legs and assemble those to person tracks (Section V-B and V-C) can, when re-trained on our large dataset, perform quite well. Their recall is limited by the methods' design, since individual legs cannot be detected at large distances. Furthermore, due to the strong assumptions regarding shape and motion, this type of method has the severe drawback of only being applicable to persons in pants; it does not readily apply to other clothing or different classes, such as wheelchairs.

Segment-classification based methods segment the scan into individual pieces which are then classified as person or background (Section V-A).

This type of method does not suffer from the aforementioned drawbacks but, even though trained on our dataset, performs significantly worse than a re-trained leg-tracking based method. Note that due to learning about persons, as opposed to legs, it can reach very high recall values.

The DROW methods are based on deeply learned voting, and do neither suffer from the principled drawbacks of the leg-tracking type, nor from the inferior performance of the segment-classification type. Specifically, the light green curve (——) shown in Figure 4 corresponds to the person detection performance of DROW *while it simultaneously detects walkers and wheelchairs*, which none of the other methods can do. This already works well with the baseline DROW3x, but the simplistic use of temporal information (without doing any tracking) further improves the results, beating current state-of-the-art by 10.9 percentage points AUC. Additionally, the red curve (——) shows the agnostic performance, meaning if the only task is to detect people and walking aids jointly, our performance even improves.

Temporal and odometry information are clearly beneficial when they can be integrated into a method. This is the case for DROW, as we have shown in Section IV, and in initial experiments with the Joint leg tracker, we saw an increase in EER of 6 percentage points when using odometry. In [11], Arras *et al*. mention that including simple motion features led to worse performance, suggesting that it is not always obvious how to make good use of temporal information. We suggest a simple but principled way of using temporal information and odometry in DROW, which shows consistent improvements, but we look forward to more research on this subject.

A larger training set further improves our results slightly (——). For this experiment, we merged the training and validation sets and use the hyperparameters selected on the validation set previously. The improved performance shows that the model has additional capacity and indicates that even better detectors can be learned given more data.

The localization accuracy of the detections can be investigated by evaluating with a smaller radius of $0.3\,\text{m}$. The corresponding curves are shown by the dashed lines in Figure 4 (·····). All leg-tracking based approaches perform significantly worse under this stricter evaluation scheme. A possible explanation for this behavior is the extra latency introduced by the leg trackers' Kalman filter[8]. Meanwhile, both the segment-classification and DROW based approaches are hardly affected. This means their detections are located more precisely, which is another clear advantage.

Hyperopt resulted in surprising parameter values for DROW. Considering all the hyperparameters of our method are very interpretable, some final values differed a lot from our initial "sane" manually selected ones. For DROW the final test performance was a little better when using our initial guesses, indicating that hyperopt might have overfitted the validation set. Nevertheless, the Leigh tracker gained about 4 percentage points AUC and, to be consistent, we report hyperopt tuned performances for all DROW curves. Since we optimized the AUC of all class curves jointly, especially the person performance typically slightly decreased. If only person detections are relevant, we could easily improve the person performance of DROW at the cost of the other classes. The fact that DROW still performs very well, even with some unintuitive values, suggests it is robust and can also easily be tuned.

Multi-target tracking systems are usually used in tandem

---

[8]It should be noted that the evaluation in [15] used a radius of $0.75\,\text{m}$, which is too lenient to be an indicator for practical use.

with detectors in practice [2], [1]. This facilitates higher-level analysis [31] and reasoning, especially across longer timeframes [2]. Nevertheless, in [27] Linder and Breuers *et al.* make the "observation that [...] detector performance is the single, most important factor influencing tracking performance which goes far beyond the impact of the chosen tracking algorithm." This is why the development of strong, multi-class detectors such as DROW is necessary for mobile robotics.

VIII. CONCLUSIONS

For mobile robot platforms that operate among people, it is a crucial ability to detect those people, be it with or without mobility aids. These robots are usually equipped with 2D laser scanners for safety reasons, making the ability to piggyback this modality for detection especially attractive. Due to the sparsity of the signal, though, this task is extremely challenging.

We release a large new set of person annotations recorded "in the wild", making this, to the best of our knowledge, the largest publicly available dataset for person detection in 2D range data. We evaluate several state-of-the-art detectors as baselines on this dataset, ready for others to compare against. (All precision-recall curves will be provided online.)

We extend the DROW detector to the task of detecting persons—in addition to, and simultaneously with wheelchairs and walkers—hereby introducing to the field an interesting departure from the classic segment-and-classify pipeline, outperforming the strongest baseline by 10.3 percentage points AUC. We solve shortcomings of the voting scheme presented in the original DROW paper, namely the discretization of detections and the inability to report meaningful confidences.

Inspired by how we as humans annotated the dataset[9], we propose initial ways of integrating temporal information into DROW, leading to substantial gains in performance, especially for the challenging person and walking aid classes. This suggests more research in this area can lead to even better detectors.

Most importantly, this work shows that DROW is the only strong, unbiased multi-class detector for 2D range data.

ACKNOWLEDGMENT

REFERENCES

[1] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, *et al.*, "SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports," in *FSR*, 2016.

[2] N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrová, J. Young, J. Wyatt, D. Hebesberger, T. Kortner, *et al.*, "The STRANDS project: Long-term autonomy in everyday environments," *RAM*, vol. 24, no. 3, pp. 146–156, 2017.

[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *CVPR*, 2012.

[4] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *CVPR*, 20167.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *ICCV*, 2017.

[6] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A Diverse Dataset for Pedestrian Detection," in *CVPR*, 2017.

[7] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition," in *IROS*, 2015.

[8] P. Ondruska, J. Dequaire, D. Zeng Wang, and I. Posner, "End-to-End Tracking and Semantic Segmentation Using Recurrent Neural Networks," in *RSS, Workshop on Limits and Potentials of Deep Learning in Robotics*, 2016.

[9] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017.

[10] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," *arXiv preprint arXiv:1711.06396*, 2017.

[11] K. O. Arras, Ó. M. Mozos, and W. Burgard, "Using Boosted Features for the detection of People in 2D Range Data," in *ICRA*, 2007.

[12] L. Spinello and R. Siegwart, "Human Detection using Multimodal and Multidimensional Features," in *ICRA*, 2008.

[13] Ch. Weinrich, T. Wengefeld, Ch. Schroeter, and H.-M. Gross, "People Detection and Distinction of their Walking Aids in 2D Laser Range Data based on Generic Distance-Invariant Features," in *RO-MAN*, 2014.

[14] C. Pantofaru, "ROS leg_detector package," https://wiki.ros.org/leg_detector, 2010.

[15] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and Following with 2D Laser Scanners," in *ICRA*, 2015.

[16] L. Beyer*, A. Hermans*, and B. Leibe, "DROW: Real-Time Deep Learning based Wheelchair Detection in 2D Range Data," *RA-L*, 2016.

[17] M. Luber, G. D. Tipaldi, and K. O. Arras, "Place-Dependent People Tracking," *IJRR*, vol. 30, no. 3, 2011.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep Into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *ICCV*, 2015, pp. 1026–1034.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ICML*, 2015.

[20] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *ICML*, 2013.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *JMLR*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *ECCV*, 2016.

[24] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," in *NIPS Deep Learning workshop*, 2013.

[26] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional Two-Stream Network Fusion for Video Action Recognition," in *CVPR*, 2016.

[27] T. Linder*, S. Breuers*, B. Leibe, and K. O. Arras, "On Multi-Modal People Tracking from Mobile Platforms in Very Crowded and Dynamic Environments," in *ICRA*, 2016.

[28] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

[29] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015.

[30] J. Bergstra, D. Yamins, and D. D. Cox, "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures," in *ICML*, 2013.

[31] L. Beyer, A. Hermans, and B. Leibe, "Biternion Nets: Continuous Head Pose Regression from Discrete Training Labels," in *GCPR*, 2015.