# Joint Operator Detection and Tracking for Person Following from Mobile Platforms

Nicolai Wojke[1], Raphael Memmesheimer[1], and Dietrich Paulus[1]

***Abstract*— In this paper, we propose an integrated system to detect and track a single operator that can switch *off* and *on* when it leaves and (re-)enters the scene. Our method is based on a set-valued Bayes-optimal state estimator that integrates RGB-D detections and image-based classification to improve tracking results in severe clutter and under long-term occlusion. The classifier is trained in two stages: First, we train a deep convolutional neural network to obtain a feature representation for person re-identification. Then, we bootstrap a classifier that discriminates the operator from remaining people on the output of the state-estimator. We evaluate the approach on a publicly available multi-target tracking dataset as well as custom datasets that are specific to our problem formulation. Experimental results suggest reliable tracking accuracy in crowded scenes and robust re-detection after long-term occlusion.**

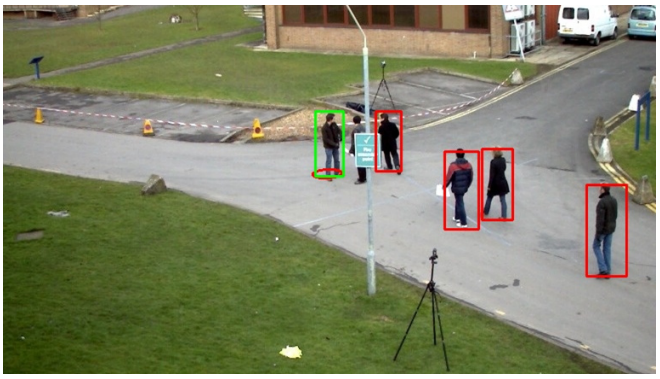

Fig. 1: Exemplary tracking output on PETS 2009 [4] sequence S2L1.

## I. INTRODUCTION

Detecting and tracking an operator through domestic environments is a key capability in human-robot interaction. The RoboCup@Home comittee has long recognized this importance and is testing operator following as a basic challenge in national and international competitions. During our participation in these events we found that, while relatively easy in controlled laboratory environments, the difficulty of the problem scales with the complexity of the application scenario. In particular when deployed in populated environments, the problem still poses scientific challenges: On the one hand, data-association uncertainty, long-term occlusions, and social interaction pose major challenges to the tracking system that often cannot be resolved based on motion information alone. At the same time, in realistic applications the operator may leave or re-enter the scene at any time, thus requiring a full re-identification from a large number of of object identities. In the end, a system to be successfully deployed under these circumstances requires careful combination of a tracking component that resolves local ambiguities and a re-identification component that recovers from tracking failures and long-term occlusions.

Due to these specific requirements, most work on people tracking is not directly applicable to our application domain. In particular, most approaches to multiple object tracking lack a re-identification component, such that they can only deal with limited-time occlusions. Further, from a computational point of view it is fundamentally more difficult to track the identity of multiple objects than tracking a single operator. Therefore, it is worth developing a system directly tailored to the application.

In this paper, we address operator following in a set-valued state estimation framework. Assuming the operator may leave or enter the field of view at any time, the target can switch between *off* and *on* at random. This leads to a state space representation in which the operator state is a set-valued random variable that is either empty or contains exactly one element. Our method is based on the Bernoulli filter [18], a specialized solution to the Bayes recursion for this particular problem domain. The filter is derived from finite set statistics, a principled theory for set-valued random variables, and is provable Bayes-optimal [18]. We combine the Bernoulli filter with a Probability Hypothesis Density filter [10] to accurately model the statistics of non-operator pedestrians in the scene. In a combined update, both filters compete for the same detections, yielding a unified framework to discriminate the operator from remaining people under consideration of data association uncertainty. A high level overview of this approach is shown in Fig. 2. The main contributions of our paper are:

- We show how to combine Bernoulli [18] and Probability Hypothesis Density [10] filter to reliably track a single person in crowded environments.
- We integrate an image-based classifier into this framework, suitable for operator re-identification after severe occlusion or when the operator leaves and re-enters the scene. The classifier is incrementally learned from the output of the tracking framework.

The remainder of this paper is organized as follows. Section II starts with a discussion of related work. In Section III

[1]Active Vision Group, Institute for Computational Visualistics, University of Koblenz-Landau, 56070 Koblenz, Germany {nwojke, raphael, paulus}@uni-koblenz.de

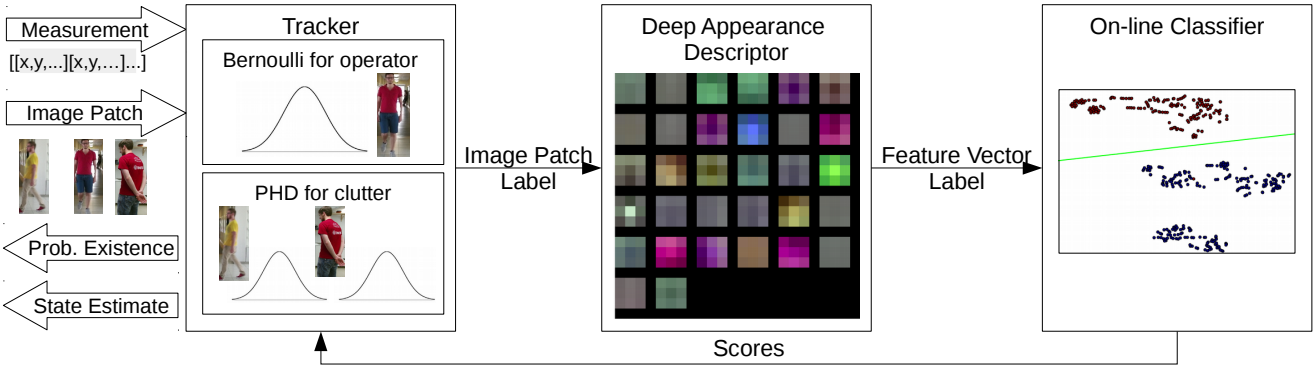


Fig. 2: Overview of our tracking approach: A combination of Bernoulli filter and Probability Hypothesis Density filter is used to track the operator. The measurement model is based on spatial information and the output of a binary classifier that discriminates the operator from remaining people in the scene.

we give a brief introduction to random finite sets. Section IV describes our approach to operator detection and tracking in a set-valued state estimation framework. In Section V we describe how we train the binary classifier that discriminates the operator from remaining people. In Section VI we present our experimental evaluation and in Section VII we give our conclusions.

## II. RELATED WORK

Few integrated systems for operator following have been presented as a whole. Leigh *et al.* [9] present a laser based people tracking approach. They detect people using extracted clusters of laser measurements at leg height. The clusters are then tracked using a combination of Kalman filter and Global Nearest-Neighbour (GNN) matching. Gockley *et al.* [5] focus on the social aspect of people following in human-computer interaction. Their tracker is a simple particle filter on laser detections.

More frequently, operator following is solved as part of a multi-target tracking system. For example, Topp and Christensen [20] design a multiple target tracker for following and passing persons. Detections are laser-based and the tracker uses a sample-based joint probabilistic data association filter. Munaro *et al.* [14] design a multiple target tracking system with GNN data association based on Kinect RGB-D detections. They first find candidate clusters in the 3-D point cloud. Then, they classify the candidates with a support vector machine that is trained on histograms of oriented gradients. They apply the system to follow people in domestic environments.

In recent revisitations, classical online multi-target tracking frameworks have shown competitive performance when compared to more recent batch algorithms. Notably, Kim *et al.* [7] show that the classical MHT algorithm [16] can achieve state-of-the-art results when combined with a visual classifier. Rezatofighi *et al.* [17] have investigated an efficient solution to the joint probabilistic data association that, combined with a heuristic track handling scheme, achieves competitive results in dense tracking scenarios with substantial occlusions, false alarms, and missed detections. These methods are, however, only capable of short-term tracking. On the other side are tracking approaches based on FISST [11], a specialized mathematical theory for set-valued random variables. Within this theoretical framework, a number of recursive state estimators have been proposed. For example, the PHD filter [10] is a moment approximation of the set-valued multi-target Bayes filter that is computationally efficient and has successfully been applied to various tracking tasks, e.g., [23]. In the case of a single target that is tracked through clutter, the Bernoulli filter [18] represents an exact solution to the Bayes recursion under consideration of data association uncertainty.

The problem formulation closest to our specific application scenario can be found in a visual object tracking context. Here, the problem is posed in an incremental learning framework that recovers from occlusions and tracking failures. For example, Hare *et al.* [6] apply a budgeted kernel support vector machine on a combination of Haar-like, histogram, and raw pixel features. They also propose to integrate this classifier into a structured prediction task to better cope with decreasing localization accuracy that is due to continuous model adaption. Nebehay *et al.* [15] explicitly address tracking articulated and deformable objects. They find matching key-points between successive frames based on descriptor similarity. Then, they group correspondences based on observed motion. While our approach is mostly in line with these methods, it tackles different aspects of the problem: Since we have RGB-D data available, our tracking system does not suffer from drift, but allows for more accurate integration of motion uncertainty. Further, since we are interested in tracking pedestrians only, we work on a specialized feature space that has been optimized for this purpose.

## III. BACKGROUND

In this section we give a brief overview of random finite sets and multi-object Bayesian filtering. For a more complete introduction to methods described here, we refer the reader to [10] and [11].

### A. Bernoulli RFS

A random finite set (RFS) is a set-valued random variable that can be described by two probability distributions: a discrete probability distribution for the cardinality of the set and a joint probability for the individual members of the set, given its cardinality. A particular class of RFS that is important to our work is the Bernoulli RFS. A Bernoulli RFS $X$ is either empty with probability $1-r$ or contains exactly one member that is distributed according to a spatial density $p(\boldsymbol{x})$. Formally, the probability density of a Bernoulli RFS can be written as

$$f(X) = \begin{cases} 1-r & X = \emptyset, \\ r \cdot p(\boldsymbol{x}) & X = \{\boldsymbol{x}\}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

With respect to our application, the state of an operator that may or may not be present in the scene can be modelled as Bernoulli RFS. In this case, $r$ is the probability that the operator is currently present and $p(\boldsymbol{x})$ is the spatial density of the operator. Further, for Bernoulli RFS there exists an exact, Bayes-optimal recursion to estimate the state from a sequence of (set-valued) measurements. The Bernoulli filter [18] incorporates clutter statistics into the recursion to estimate the probability of existence as well as the spatial density in light of data association uncertainty.

### B. Probability Hypothesis Density

For RFS that contain more than one element, the exact Bayes recursion is computationally intractable, but approximations exist. For example, the Multi-Target Multi-Bernoulli (MeMBer) recursion [21] can be employed to estimate the density of multiple independent Bernoulli RFS. As an alternative, one may resort to estimating statistical moment descriptors, instead. The first-order moment of an RFS $X$ is a non-negative function $v(\boldsymbol{x})$ which integrates to the expected number of elements. This function is called the Probability Hypothesis Density (PHD) or *intensity*. With respect to our work it is worth noting that the PHD of a Bernoulli RFS can be computed exactly by $v(\boldsymbol{x}) = r \cdot p(\boldsymbol{x})$ and that it is possible to convert between the PHD and the probability density of a Bernoulli RFS without loss of information. For other types of RFS, the PHD is only a summary descriptor that is often easier to compute.

## IV. JOINT DETECTION AND TRACKING

In this section, we describe our integrated filtering framework for joint operator detection and tracking. For simplicity, we assume that we are given a binary classifier that discriminates the operator from all other people in the environment. Information on how we learn this classifier are given in Section V.

### A. Overview

We assume a mobile robot collects measurements of pedestrians in the field of view using a standard pedestrian detector, e.g., from leg segmentations in laser scans [1] or from 3D point clusters obtained by a Microsoft Kinect [13]. We assume that these detections are collected from a sensor that is calibrated against a color camera to perform image-based operator classification. Further, we assume that the operator may or may not be currently visible. Therefore, the operator target can switch *on* and *off*.

Since a standard pedestrian detector reports not only the operator location, but also the location of other pedestrians in the field of view, a critical part of the tracking framework is to identify the correct detection to use for updating the operator state. Here, we follow a RFS methodology to avoid this hard decision and, instead, integrate all measurements into the estimation framework. In particular, we formulate our problem on an augmented state space that contains the kinematic state and a binary object class label that identifies the operator. Then, we apply a multi-object filter recursion to estimate the state from all measurements up to the current time step. This is fundamentally different from traditional single-target tracking in that hard decisions are avoided and data association uncertainty is explicitly handled by the state estimator. Further, by integrating information about non-operators we render the clutter model of the operator process more accurate.

### B. Set-valued State Estimation

Following a RFS methodology, we model the set of objects at time $k$ as finite set-valued random variable

$$X_k = X_k^{(0)} \cup X_k^{(1)} \tag{2}$$

for which we explicitly distinguish the operator set $X_k^{(1)}$ from all non-operators in $X_k^{(0)}$. This distinction is manifested by tracking on an augmented state space $\boldsymbol{w} = (\boldsymbol{x}, \beta)$ that contains the kinematic state $\boldsymbol{x}$, in our application ground plane position and velocity, and a binary object class label $\beta \in \{0, 1\}$ that evaluates to 1 for the operator in $X_k^{(1)}$ and 0 for all non-operator objects in $X_k^{(0)}$. Based on this formalization one can choose from various set-valued filtering frameworks to estimate the statistics of multi-object state $X_k$ from a sequence of (set-valued) measurements. For example, one could apply a PHD filter [10] to estimate the first-order moment descriptor or a MeMBer filter [21] to recover the probability density under the assumption that $X_k$ is a union of independent Bernoulli RFS. However, given that we know that $X_k^{(0)}$ is Bernoulli, i.e., contains at most one object, and given that we do not need to resolve object identities for any non-operator object, we resort to a combination of the exact Bernoulli filter for $X_k^{(1)}$ and a computationally efficient PHD filter for remaining objects in $X_k^{(0)}$. This idea is visualized in Fig. 3. At each time step we obtain the PHD of non-operators $v_{k|k}^{(0)}(\boldsymbol{x})$ as well as the probability density $f_{k|k}^{(1)}(X)$ of the Bernoulli operator RFS. Assuming statistical independence, we perform independent prediction steps. However, both processes compete for the same measurements in a combined update. This is elaborated in more detail in the following sections.

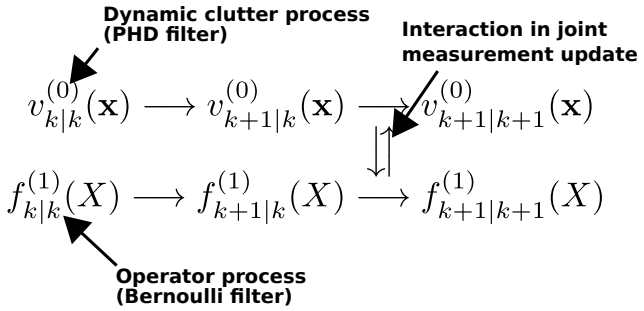


Fig. 3: Conceptual overview of the tracking framework: The probability distribution of the operator RFS is tracked using a Bernoulli filter. A PHD filter estimates the statistics of the non-operator process. In a combined update, both processes compete for the same detections.

### C. Prediction

Assuming that the operator identity does not change during tracking, the motion model on the augmented state space simplifies to

$$p(\boldsymbol{x}_{k+1}, \beta_{k+1} \mid \boldsymbol{x}_k, \beta_k) = \begin{cases} p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k), & \beta_{k+1} = \beta_k, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k)$ is a suitable motion model that describes how the kinematic state of a single pedestrian evolves over time. In our particular application, we apply a constant velocity motion model. Note that, since objects do not change their class label, we can apply separate predictions for non-operators in $X_k^{(0)}$ and the operator $X_k^{(1)}$. Consequently, we perform the standard PHD filter [10] and Bernoulli filter [18] prediction with no modification or interaction between the two.

### D. Update

At each time step $k$ we obtain a new measurement set $Z_k = \{\boldsymbol{z}_{k,1}, \ldots, \boldsymbol{z}_{k,N_k}\}$, where each measurement vector $\boldsymbol{z}_{k,i} = (x_{k,i}, y_{k,i}, s_{k,i})$ contains a position and a classification confidence score. More precisely, this score is the signed distance to the decision function of our binary operator classifier. Given these measurements, we factorize the measurement model into a spatial likelihood conditional on the kinematic state and a classification likelihood conditional on object class:

$$p(\boldsymbol{z}_{k,i} \mid \boldsymbol{x}_k, \beta_k) = p(x_{k,i}, y_{k,i}, s_{k,i} \mid \boldsymbol{x}_k, \beta_k), \tag{4}$$

$$= p(x_{k,i}, y_{k,i} \mid \boldsymbol{x}_t) p(s_{k,i} \mid \beta_k). \tag{5}$$

That is, we assume that the measurement generation process of the spatial component is identical for operator and non-operators and we assume that appearance is independent of object location. Note that this model incorporates spatial information that are particularly useful to discriminate between objects when uncertainty is low as well as appearance information that remain valid over longer periods of time, e.g., to re-detect an operator after long-term occlusions or on re-entry into the scene. Thus, due to integrating a classifier into the measurement model we combine tracking and re-detection into a unified filtering framework.

In our combined update we proceed as follows: With respective to the update equations of the PHD and Bernoulli filter, non-operators present themselves as clutter to the operator process. Similarly, the operator is clutter with respect to non-operators. Consequently, we substitute the intensity of the predicted non-operator RFS for the clutter intensity in the Bernoulli filter update. Similarly, we substitute the intensity of the predicted operator RFS for the clutter intensity in the PHD filter update. Conceptually, this is identical to the augmented PHD update in [12]. We only exchange the PHD filter of the foreground process by the exact filter recursion for Bernoulli RFS to obtain more accurate results. Due to space limitations we refer to [10] and [18] for individual filter update equations and provide the combined filter update in the appendix of this paper.

TABLE I: CNN architecture of the deep feature space [22]

| Name | Patch Size/Stride | Output Size |
|---|---|---|
| Conv 1 | $3 \times 3/1$ | $32 \times 128 \times 64$ |
| Conv 2 | $3 \times 3/1$ | $32 \times 128 \times 64$ |
| Max Pool 3 | $3 \times 3/2$ | $32 \times 64 \times 32$ |
| Residual 4 | $3 \times 3/1$ | $32 \times 64 \times 32$ |
| Residual 5 | $3 \times 3/1$ | $32 \times 64 \times 32$ |
| Residual 6 | $3 \times 3/2$ | $64 \times 32 \times 16$ |
| Residual 7 | $3 \times 3/1$ | $64 \times 32 \times 16$ |
| Residual 8 | $3 \times 3/2$ | $128 \times 16 \times 8$ |
| Residual 9 | $3 \times 3/1$ | $128 \times 16 \times 8$ |
| Dense 10 | | 128 |
| Batch and $\ell_2$ normalization | | 128 |

## V. OPERATOR CLASSIFICATION

A key property of our filtering framework is the ability to re-identify the operator on re-entry and after long-term occlusions. This is due to integration of an image-based classifier that discriminates the operator from all other people in the environment. In practice, the performance of our tracking framework largely depends on the accuracy of this classifier, because when the operator is occluded or abscent from the scene, state estimation uncertainty can grow to a level where it is not possible to identify the operator purely based on its estimated position. In order to learn a well-discriminating classifier we proceed in a two-step approach. During online application, we bootstrap a classifier for the specific operator on training examples obtained from tracking. This classifier operates on a feature space that has been trained to discriminate people in a person re-identification setting, prior to the operator following application.

### A. Deep Appearance Descriptor

Our image-based appearance descriptor is based on a deep neural network [22] that has been trained on a large-scale person re-identification dataset [25] with over 1,100,000 images of 1,261 pedestrians. The network architecture was designed with a people tracking application in mind and pre-trained models are available from the authors' GitHub

repository (see [22] for more information on the original tracking application). An overview of the network architecture is given in Table I. In summary, it is a wide residual network [24] with two convolutional layers followed by six residual blocks. The global feauture map of dimensionality 128 is computed in dense layer 10. A final batch and $\ell_2$ normalization projects features onto the unit hypersphere. In total, the network has 2,800,864 parameters and one forward pass of 32 bounding boxes takes approximately 30 ms on an Nvidia GeForce GTX 1050 mobile GPU. Thus, this network is well suited for online tracking, provided that a modern GPU is available.

### B. Online Learning

For online operator following, we train a linear classifier on top of the deep feature space and model the classification likelihood in our measurement model (5) by two Gaussian distributions which we place on each side of the decision plane. At each time step, the classifier is trained on the newly arrived detections and a random selection of detections from previous times using stochastic gradient descent. As positive example we take the detection that corresponds to the most likely operator location. Remaining detections are used as negative examples. Further, we only train the classifier when the probability that the operator is present is larger than a predefined threshold. Thereby, we avoid learning when the operator is currently not visible. We also only learn on image patches that are occluded by at most $30\%$ to avoid learning mixed identities.

## VI. EXPERIMENTS

Existing people tracking datasets have either been created for evaluation of multi-target tracking systems (e.g., [4], [8]) or do not contain a combination of depth/laser and image data. To the best of our knowledge, there exists no publicly available dataset that targets the specific task of tracking a single operator and contains all the required sensor data. To evaluate all the specific components of our system under these circumstances we proceed as follows. First, we evaluate our system on a widely accepted multi-target dataset [4] by tracking single individuals from a given starting position. Then, we apply our tracker to a custom dataset with common pitfalls specific to single-operator tracking, in particular operator disappearance and re-entry. These situations are not commonly modelled in multi-target datasets. To establish a comparison against the current state of the art, we compare our method against two well-established visual object tracking systems, namely CMT [15] and Struck [6]. Performance is measured in terms of tracking accuracy according to a selection of CLEAR MOT [3] metrics: (1) A precision score indicates the fraction of operator locations that coincide with the ground truth location, (2) a recall score indicates the fraction of ground truth locations that have been reported by the tracker. In addition we count the number of identity switches, that is the number of times that the tracker switches between a ground truth object identity.

TABLE II: Results on PETS 2009 S2L1 [4]

| | **Precision** | **Recall** | **ID** |
|---|---|---|---|
| Identity 1 | | | |
| Ours (motion & appearance) | **1.0000** | **0.8984** | **0** |
| Ours (motion only) | 0.6056 | 0.4518 | 3 |
| CMT [15] | 0.2526 | 0.2522 | 12 |
| Struck [6] | 0.9212 | 0.9212 | 5 |
| Identity 3 | | | |
| Ours (motion & appearance) | **1.0000** | **0.7444** | **0** |
| Ours (motion only) | 0.1473 | 0.1241 | **0** |
| CMT [15] | 0.5434 | 0.5414 | 1 |
| Struck [6] | 0.2105 | 0.2105 | 2 |
| Identity 7 | | | |
| Ours (motion & appearance) | **1.0000** | 0.9390 | 0 |
| Ours (motion only) | **1.0000** | 0.8537 | 0 |
| CMT [15] | **1.0000** | 0.0122 | 2 |
| Struck [6] | **1.0000** | **1.0000** | **0** |
| Identity 9 | | | |
| Ours (motion & appearance) | **1.0000** | **0.8243** | **0** |
| Ours (motion only) | 0.9891 | 0.6988 | **0** |
| CMT [15] | 0.1699 | 0.1699 | 8 |
| Struck [6] | 0.6795 | 0.6795 | 8 |

### A. Parametrization

Our implementation is based on a Gaussian mixture implementation of the Bernoulli and PHD filter. In all of our experiments we adopt a single set of parameters. In particular, the constant velocity motion model adds isotropic noise with standard deviation $\Delta t \cdot 1\,\mathrm{m}$ for the position and $\Delta t \cdot 1\,\mathrm{m/s}$ for the velocity, where $\Delta t$ is the time gap between consecutive frames. The spatial component of the measurement model adds isotropic noise with standard deviation $0.1\,\mathrm{m}$. In addition, we use the following set of parameters for the Bernoulli and PHD filter: The probability of survival is set to $0.95$ and the probability of detection is set to $0.7$. We employ a partial uniform birth model for the clutter PHD filter [2] with clutter intensity $3 \cdot 10^{-2}$ and a single Gaussian component for operator birth at the known location of first appearance. In addition, we clamp the probability of existence of the operator state at $10^{-4}$ to prevent this value to become indefinitely small when the operator leaves the field of view. To measure the impact of image-based classification on overall results, we run the experiment once with and once without image-based classifier. With classification enabled, the class-conditional appearance likelihood is modelled as normal distribution around the class label, i.e., $\mathcal{N}(-1, 1^2)$ and $\mathcal{N}(1, 1^2)$. Without image-based classifier, we only evaluate the spatial component of the measurement model.

### B. PETS 2009

In the first experiment, we apply our tracker to sequence S2L1 of the PETS 2009 [4] dataset. This sequence is only moderately crowded, but contains complex interactions and occlusions against static scene geometry (c.f. Fig. 1). Bounding box detections and ground truth have been taken from the MOTChallenge website[1]. The world-space position of each detection is computed from known camera calibration.

[1] http://www.motchallenge.net

TABLE III: Results on custom datasets

| | Precisior | Recall | ID |
|---|---|---|---|
| Sequence 1 | | | |
| Ours (motion & appearance) | **1.0000** | 0.6083 | **0** |
| Ours (motion only) | 0.7761 | 0.4551 | 5 |
| CMT [15] | 0.6849 | 0.6849 | – |
| Struck [6] | 0.7112 | **0.7112** | – |
| Sequence 2 | | | |
| Ours (motion & appearance) | **1.0000** | **0.7411** | **0** |
| Ours (motion only) | 0.9805 | 0.6286 | 2 |
| CMT [15] | 0.6518 | 0.6518 | – |
| Struck [6] | 0.6196 | 0.6196 | – |
| Sequence 3 | | | |
| Ours (motion & appearance) | **0.9097** | **0.6168** | **3** |
| Ours (motion only) | 0.6878 | 0.3447 | 6 |
| CMT [15] | 0.3787 | 0.3787 | – |
| Struck [6] | 0.1837 | 0.1837 | – |

Fig. 4: In sequence 3 of our custom dataset similar clothing stresses the re-identification component of the tracking framework.

Individuals that we have selected for the purpose of evaluation have been chosen to reflect a variety of appearances and difficulty levels. The results are summarized in Table II. In general, our approach performs favorable compared to CMT and Struck. On all sequences we obtain the highest precision and fewest ID switches. In particular, due to integration of the image-based classifier we successfully discriminate the operator from remaining pedestrians, thus reaching a $100\%$ precision score and $0$ identity switches. Without image-based classifier (motion only), the tracker fails to pick up the correct object identity after long-term occlusion against static scene geometry (identity 3). In this case, motion uncertainty raises to a level where the operator cannot be identified purely based on its position. Struck and CMT generally perform considerably worse. More specifically, Struck successfully tracks the operator in two out of the four evaluation scenarios (identity 1 and 7). CMT generally looses the operator early on during tracking, as indicated by low recall rates. In general, the lower performance of Struck and CMT can be attributed to the more general tracking application they have been designed for: Whereas the operator classifier in our tracking framework is trained on positive and negative detections throughout the online tracking application, Struck and CMT collect negative training examples from the entire image, thus yielding a less informed model for the specific operator following task.

### C. Custom Datasets

In a second experiment we evaluate the performance on a custom dataset that is specific to operator tracking. Therefore, these sequences exhibit long-term occlusions and operator disappearances to stress the re-detection component of our tracking system. The dataset has been collected from a domestic service robot that is equipped with a Microsoft Kinect 2. Detections have been generated using a people detector [13] that is available in current releases of the *Point Cloud Library* [19]. This detector generates point clusters on the ground plane which are subsequently filtered in multiple iterations. Then, a HOG classifier decides whether the particular cluster displays a person. In total, we have collected three sequences of different complexity:

- Sequence 1: In this sequence the operator walks away from the robot; several people cross the path to block the view.
- Sequence 2: This sequence is similar to the first, but two identities wear identical shirts.
- Sequence 3: The most challenging sequence exhibits complex motion, interaction, and longer periods of operator disappearance. In addition, all identities are similarly dressed (c.f. Fig. 4).

The ground truth operator position is obtained from a motion capturing system that is built of 12 OptiTrack Prime 13 cameras. These cameras record the 3D position of the operator and the robot position at 120 Hz. For evaluation, detections and ground truth positions have been projected into the RGB-D camera frame. The results of our evaluation are summarized in Table III. Again, Struck and CMT perform considerably worse than the Bernoulli tracker. In particular, we found both approaches struggle with sudden appearance changes, e.g., due to the operator turning away from the camera in the beginning of the sequence. The proposed Bernoulli tracker on the other hand reliably tracks the operator throughout most of the sequences. The only tracking failure we observed during our evaluation happened in the third, most challenging sequence. In this case, all identities are similarly dressed and discrimination based on object appearance alone is challenging (c.f. Fig. 4). Consequently, during a period of longer absence from the scene the Bernoulli tracker picks up a wrong identity twice before the operator is re-detected. For qualitative inspection we provide videos of the tracking output on our project page[2].

## VII. CONCLUSIONS

We have presented an integrated approach to jointly detect and track a single operator in crowded and cluttered environments. A key property of our framework is the integration of spatial measurements and an image-based classifier in a way that enables operator re-detection after long-term occlusions or absence from the scene. The framework is built around

[2]https://userpages.uni-koblenz.de/~nwojke/bernoulli_tracker

a set-valued, Bayes-optimal state estimator that provides a mathematically elegant representation of the operator state as set-valued random variable. In our experimental evaluation we found the method works well in many practical application scenarios.

## APPENDIX

In this section we detail our combined Bernoulli/PHD filter update. At each time step $k$ we are given the predicted intensity of non-operators $v^{(0)}_{k|k-1}(\boldsymbol{x})$ as well as the probability density of the predicted operator state $f^{(1)}_{k|k-1}(X)$ which is parametrized by the probability of existence $r_{k|k-1}$ and spatial density $p_{k|k-1}(\boldsymbol{x})$. As outlined in the paper, we can compute the intensity of the operator RFS by

$$v^{(1)}_{k|k-1}(\boldsymbol{x}) = r_{k|k-1} p_{k|k-1}(\boldsymbol{x}). \tag{6}$$

This computation is exact and without loss of information.

### A. PHD Update

In order to update the intensity of non-operators we write down the PHD update equation on the augmented state space. Then, we substitute (6) for the intensity of the operator. According to [10], the posterior intensity on the augmented state space is

$$v_k(\boldsymbol{x}, \beta) = [1 - p_{\mathrm{D}}(\boldsymbol{x})]\, v_{k|k-1}(\boldsymbol{x}, \beta) + \sum_{\boldsymbol{z} \in Z_k} v_k(\boldsymbol{z}, \boldsymbol{x}, \beta) \tag{7}$$

with

$$v_k(\boldsymbol{z}, \boldsymbol{x}, \beta) = \frac{p_{\mathrm{D}}\, p(\boldsymbol{z} \mid \boldsymbol{x}, \beta) v_{k|k-1}(\boldsymbol{x}, \beta)}{c_k(\boldsymbol{z}) + \langle p_{\mathrm{D}}\, p(\boldsymbol{z} \mid \cdot, \beta), v_{k|k-1} \rangle}. \tag{8}$$

Above, $p_{\mathrm{D}}(\boldsymbol{x})$ is the probability of detection independent of object class, $p(\boldsymbol{z} \mid \boldsymbol{x}, \beta)$ is the measurement model (c.f., Eq. 5), and $c_k(\boldsymbol{z})$ is the clutter intensity that characterizes false alarms. We also use the following short-hand notation for the inner product: $\langle f(\cdot), p \rangle = \int f(\boldsymbol{x}) p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$. Now, abbreviating $v^{(\beta)}(\boldsymbol{x}) = v(\boldsymbol{x}, \beta)$ we get

$$v^{(\beta)}_k(\boldsymbol{x}) = [1 - p_{\mathrm{D}}(\boldsymbol{x})]\, v^{(\beta)}_{k|k-1}(\boldsymbol{x}) + \sum_{\boldsymbol{z} \in Z_k} v^{(\beta)}_k(\boldsymbol{z}, \boldsymbol{x}) \tag{9}$$

with

$$v^{(\beta)}_k(\boldsymbol{z}, \boldsymbol{x}) = \frac{p_{\mathrm{D}}(\boldsymbol{x})\, p(\boldsymbol{z} \mid \boldsymbol{x}, \beta) v^{(\beta)}_{k|k-1}(\boldsymbol{x})}{c_k(\boldsymbol{z}) + \sum_{\eta=1}^{2} \langle p_{\mathrm{D}}\, p(\boldsymbol{z} \mid \cdot, \eta), v^{(\eta)}_{k|k-1} \rangle}. \tag{10}$$

For $\beta = 0$ this update computes the posterior intensity of the non-operator RFS. Intuitively, the operator presents itself to non-operators as additional clutter source, such that measurements that are more likely generated by the operator have little influence on the cardinality of the non-operator RFS.

### B. Bernoulli Update

While we could use the PHD update equation from the previous section to compute the intensity of the Bernoulli operator RFS by evaluating (9) for $\beta = 1$, we apply the Bernoulli filter recursion, instead. Using the Bernoulli filter update we gain additional accuracy, because it implements the exact multi-object Bayes recursion instead of an approximation [18].

Let $\kappa_k(\boldsymbol{z})$ denote the clutter intensity of the Bernoulli filter. Then, according to [18], the Bernoulli filter update is:

$$r_{k|k} = \frac{1 - \Delta_k}{1 - r_{k|k} \Delta_k}\, r_{k|k-1}, \tag{11}$$

$$p_{k|k}(\boldsymbol{x}) = \frac{1 - p_{\mathrm{D}}(\boldsymbol{x}) + p_{\mathrm{D}}(\boldsymbol{x}) \sum_{\boldsymbol{z} \in Z_k} \frac{p(\boldsymbol{z}|\boldsymbol{x})}{\kappa_k(\boldsymbol{z})}}{1 - \Delta_k} p_{k|k}(\boldsymbol{x}), \tag{12}$$

with

$$\Delta_k = \langle p_{\mathrm{D}}, p_{k|k-1} \rangle - \sum_{\boldsymbol{z} \in Z_k} \frac{\langle p_{\mathrm{D}}\, p(\boldsymbol{z} \mid \cdot), p_{k|k} \rangle}{\kappa_k(\boldsymbol{z})}. \tag{13}$$

Following the same idea as in the previous section, the false alarm RFS of the Bernoulli filter contains clutter and detections due to non-operators, such that

$$\kappa_k(\boldsymbol{z}) = c_k(\boldsymbol{z}) + \langle p_{\mathrm{D}}\, p(\boldsymbol{z} \mid \cdot), v^{(0)}_{k|k-1} \rangle \tag{14}$$

denotes the intensity of clutter RFS.

## ACKNOWLEDGMENT

We thank the Institute for Systems and Robotics at Instituto Superior Técnico, U. Lisboa, Portugal, for letting us use their motion capturing system and helping us in acquiring the ground truth data.

## REFERENCES

[1] K. O. Arras, O. M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *ICRA*, pages 3402–3407, 2007.

[2] M. Beard, B.-T. Vo, B.-N. Vo, and S. Arulampalam. Gaussian mixture PHD and CPHD filtering with partially uniform target birth. In *FUSION*, pages 535–541, 2012.

[3] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP J. Image Video Process*, 2008, 2008.

[4] J Ferryman and A Shahrokni. An overview of the PETS 2009 challenge. 2009.

[5] R. Gockley, J. Forlizzi, and R. G. Simmons. Natural person-following behavior for social robots. In *SIGCHI/SIGART*, pages 17–24, 2007.

[6] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L. Hicks, and Philip H. S. Torr. Struck: Structured output tracking with kernels. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2096–2109, 2016.

[7] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. Multiple hypothesis tracking revisited. In *ICCV*, pages 4696–4704, 2015.

[8] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, 2015.

[9] A. Leigh, J. Pineau, N. A. Olmedo, and H. Zhang. Person tracking and following with 2d laser scanners. In *ICRA*, pages 726–733, 2015.

[10] R. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Trans. Aerosp. Electron. Syst.*, 39(4):1152–1178, 2003.

[11] R. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Norwood, MA, USA, 2007.

[12] R. Mahler, B.-T. Vo, and B.-N. Vo. CPHD filtering with unknown clutter rate and detection profile. *IEEE Trans. Signal Process.*, 59(8):3497–3513, 2011.

[13] M. Munaro, F. Basso, and E. Menegatti. Tracking people within groups with RGB-D data. In *IROS*, pages 2101–2107, 2012.

[14] M. Munaro and E. Menegatti. Fast RGB-D people tracking for service robots. *Auton. Robots*, 37(3):227–242, 2014.

[15] Georg Nebehay and Roman P. Pflugfelder. Consensus-based matching and tracking of keypoints for object tracking. In *IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, March 24-26, 2014*, pages 862–869. IEEE Computer Society, 2014.

[16] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control*, 24(6):843–854, 1979.

[17] S.H. Rezatofighi, A. Milan, Z. Zhang, Qi. Shi, An. Dick, and I. Reid. Joint probabilistic data association revisited. In *ICCV*, pages 3047–3055, 2015.

[18] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina. A tutorial on bernoulli filters: Theory, implementation and applications. *IEEE Trans. Signal Process.*, 61(13):3406–3430, 2013.

[19] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA*, pages 1–4, 2011.

[20] E. A. Topp and H. I. Christensen. Tracking for following and passing persons. In *IROS*, pages 2321–2327, 2005.

[21] B.-T. Vo, B.-N. Vo, and A. Cantoni. The cardinality balanced multi-target multi-bernoulli filter and its implementations. *IEEE Trans. Signal Process.*, 57(2):409–423, 2009.

[22] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. *arXiv:1703.07402 [cs.CV]*, 2017.

[23] Nicolai Wojke and Dietrich Paulus. Global data association for the probability hypothesis density filter using network flows. In *ICRA*, pages 567–572, 2016.

[24] S. Zagoruyko and N. Komodakis. Wide residual networks. pages 1–12, 2016.

[25] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian. MARS: A video benchmark for large-scale person re-identification. In *ECCV*, 2016.