in: Proc. 6th European Conference on Mobile Robots (ECMR 2013), Barcelona, Spain

Multi-Modal People Tracking on a Mobile Companion Robot

Michael Volkhardt, Christoph Weinrich, and Horst-Michael Gross[1]

Abstract— **People detection and tracking are key aspects in current research on mobile robots. While plenty of research is focused on pedestrian tracking in public areas, fewer work exists on practical people tracking in home environments with non static cameras. This paper presents a real-time people tracking system for mobile robots that applies multiple asynchronous detection modules and an efficient Kalman filter. It allows for upright pose – and under restrictions, sitting pose – people tracking in home environments. We evaluate the performance of the tracking system using different detection modalities on newly collected indoor data sets. These data sets are made publicly available for comparison and benchmarking.**

I. INTRODUCTION

A long-term research goal is the development of mobile robots assisting users in domestic environments. Helping elderly people to live independently for as long as possible by supporting them in their daily routine and increasing their quality of life could become a major challenge in modern society. Mobile robots can add an additional benefit to the solution of this challenge by providing services that cannot be done by human care-givers – either due to time or cost restrictions. To provide these user-centered services, the robot needs to be aware of the user's position in the apartment. While a lot of current research projects focus on the detection and tracking of pedestrians, fewer works put an emphasis on people tracking in home environments. Yet, home environments introduce new challenges like partial occlusions, various poses of the user, and limited computational resources of the mobile platform that are worth exploring [1]. Additionally, most data sets used in former works cover pedestrians in outdoor scenarios or only contain images of static indoor cameras. Only a few public indoor data sets exist that are captured by a mobile robot and provide multi-modal sensor cues, like images and range data [2]. While the data set of [2] is very large and contains various sensor modalities, it does not contain a global robot position with uncertainties and labeled person IDs which are both useful to evaluate tracking algorithms for mobile robots.

Therefore, in this paper, we present an indoor data set recorded on our mobile robot platform containing data of multiple sensors – fisheye images, 3D range data (Kinect sensor), and 2D range data (laser range finder) – and additional data of the mobile robot. Furthermore, as a main contribution of this paper, we present a people tracking system that fuses detections of multiple asynchronously working detection modules while respecting the uncertainties of the different sensor cues and the pose of the robot. We evaluate the usefulness of different detection methods on the data sets by comparing the tracking capabilities of the system using different combinations of input cues. A practical solution of the tracking system, running in real-time on the robot's hardware (the robot and its architecture is described in [1]), does not include all modules and applies a trade-off between detection rate and computational performance to allow for user interaction while keeping enough CPU time for other required modules of the robot, e.g. localization and path planning. As performance is not totally satisfying in all scenarios, we show which state-of-the-art detection methods would improve the tracking on the robot the most.

The remainder of this paper is organized as follows: Section II summarizes related work in the research area. Section III presents our tracking system. Section IV describes the data sets used for evaluation and the results of our experiments. Sec. V summarizes our contribution and gives an outlook.

II. RELATED WORK

People detection and tracking are well-established research areas, and impressive results have been accomplished recently. A plenty amount of visual detection methods originated in the field of pedestrian detection, each with their own benefits and disadvantages (a survey is given in [3]). Recent approaches like [4], [5] achieve good results at frame-rate by applying a soft-cascade, tuning features, sampling the image pyramid and using ground plane constraints. Yet, pedestrian detection only covers a part of the problem of finding people in their homes, e.g. related to the variety of poses encountered and occlusions. On the other hand, [6], [7] are designed for detection quality and achieve impressive results given partial occlusion and varying poses. Unfortunately, they require several seconds of processing time per image. In the field of mobile service robotics, people are also often detected by their faces [8], color [9], and gradient features [10]. Additionally, most mobile robots are equipped with laser range finders which allow the detection of human legs [11]. Plenty of research has been done to develop methods for people tracking on mobile robots in real-world applications. Most of these approaches focus on pedestrian tracking [12], [13], [14]. Furthermore, evaluation is often done on pre-captured data, and real-time performance retreats into the background while the main focus concentrates on detection quality. On the other hand, real-time approaches usually apply very fast detectors [4], [5], a tracking-by-detection

[1]This work has received funding from the Federal State of Thuringia and the European Social Fund (OP 2007-2013) under grant agreement N501/2009 to the project SERROGA (project number 2011FGR0107). All authors are with Neuroinformatics and Cognitive Robotics Lab, Ilmenau University of Technology, 98684 Ilmenau, Germany michael.volkhardt@tu-ilmenau.de

scheme [15] and special hardware, like stereo-cameras and dedicated GPUs, which are unfortunately not available on our mobile robot platform [1]. While GPUs can greatly increase computational performance, they consume a lot of energy and heavily decrease the operational time of mobile robots. Real-time indoor approaches use thermal cameras [10] or focus on single poses and person recognition [9]. Unfortunately, they work on closed data sets, which makes comparison hard. Furthermore, many approaches do not consider the processing time for other required modules like Monte Carlo localization (MCL), Simultaneous Localization and Mapping (SLAM), or path planning. The tracking system presented in this paper runs on a single CPU while keeping enough processing time for localization, navigation, and user interaction.

III. MULTI-MODAL PEOPLE TRACKING

In the following, we describe the detection modules and the alignment of their detections, followed by a description of the tracking system.

A. Person Detection

Our real-time set-up of the people tracker uses well established methods, like Histogram of Oriented Gradients (HOG), motion, face, and leg detection, whose detection quality is mediocre compared to cutting-edge methods of Sec. II. Yet, the people tracker is also evaluated offline on our data sets applying promising new detection paradigms, i.e. [4], [6], which are not yet useable on our robot, but could be integrated in the future.

1) HOG Detection: To detect people by their body shape, we apply a full body and an upper body shape detector based on Histograms of Oriented Gradients [16], [17]. We use a scale factor between two layers of the image pyramid of 1.1 for performance reasons. A ground plane constraint for sitting and standing people is used to reduce false positives. This also increases the processing performance by a factor of 2 compared to processing the full image.

2) Face Detection: The face detection system utilizes the well-known AdaBoost detector of Viola & Jones [8]. The method is configured to detect faces up to a minimum size of 30*x*30 pixels with a scale change between two pyramid levels of 1.1. We apply the detector only on the upper half of the image to reduce processing time and false positives.

3) Motion Detection: Each time the robot does not move, which is signaled by the robot's odometry, we apply a simple motion difference detection. The difference image between two frames is thresholded, and a connected components algorithm gives bounding boxes of moving regions in the image.

4) Leg detection: The leg detection module uses range data delivered by the robot's laser range finder (LRF) and applies a boosted set of classifiers to distinguish legs from other objects in the environment [11]. By searching for paired legs, the system produces hypotheses of the user's position. However, objects similar to legs, like tables and chairs, often lead to false positive detections.

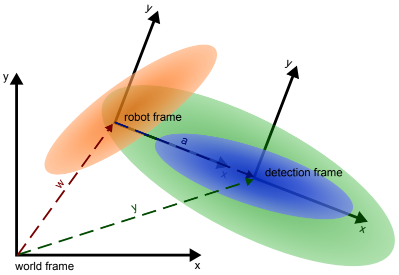

Fig. 1. The graphic shows the covariance of the concatenated transformation **y** (green) of the two uncertain transformations **w** (orange) and **a** (blue). Hence, the uncertainty of the robot's pose and the detections is propagated to the uncertainty of the detections in the world frame.

5) Fastest Pedestrian Detector in the West (FPDW): To show how our system would improve with a state-of-the-art pedestrian detection method, we applied the Matlab implementation method of [4] offline on the images of the captured data sets, transformed the bounding boxes into Gaussians, and integrated them into the people tracker.

6) Part HOG: We reimplemented the method of [6] in a multi-core C++ version which increases the perfomance compared to the Matlab version by a factor of 2. Nevertheless, the method still requires 3 seconds to process a 640*x*480 image when using a VOC 2009 model [18] and could only be evaluated offline.

B. Alignment and Transformation of Detections

Each detection module detects people by different body parts, e.g. the face, legs, or head-shoulder contour. To facilitate fusion in the people tracker, we transform the detections to Gaussians in a world coordinate frame and align them to a common reference point, i.e. the head of a person. The vision modules produce bounding boxes which are first transformed into Gaussian distributions in the camera coordinate frame using the intrinsic parameters of the camera. The distance to the robot's camera is estimated by assuming a detector specific, empirically determined metric width of the bounding boxes. These Gaussians are then transformed into world coordinates (world frame) by using the extrinsic parameters of the camera and the robot's pose. The leg detection module generates Gaussian distributions in the laser scanner's coordinate frame with x,y given by the position of the detection and the height z set to zero. These are transformed into the world coordinate frame, too. The sensor model describes the certainty of each detector and is incorporated into the covariance of the corresponding Gaussian distribution. We use an overall low variance for leg detections but a high variance in the view-direction of the robot's camera for visual detection modules, because the distance to the robot is estimated by the width of the bounding boxes which usually have a high deviation.

Compared to image based trackers, tracking in a world frame includes the motion of the robot and facilitates tracking by allowing to apply a linear motion model. One key idea of our approach is that the transformation of detections from the local sensor frames into the world frame needs to respect

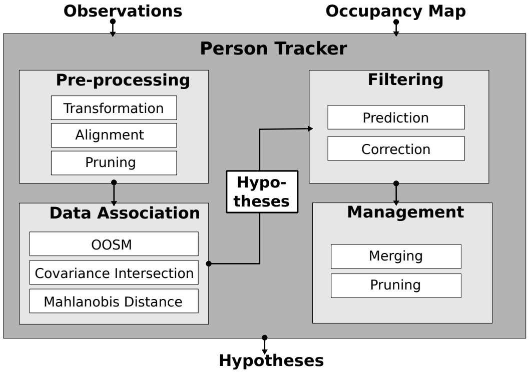

Fig. 2. Overview of the processing steps of the people tracker.

the uncertainty of the robot's pose given by the MCL. Since the pose of the robot is usually uncertain in the world frame, the covariances of the Gaussians in the local sensor frames must be increased by this uncertainty. This is visualized in Fig. 1: transformation $\mathbf{w}$ denotes the robot's pose in the world frame with an uncertainty represented by the orange Gaussian. A detection with high variance in distance estimation (camera is looking in x-direction of the robot frame) is defined by a transformation $\mathbf{a}$ that describes its position and uncertainty (blue Gaussian) in the robot frame. The covariance of the detection in the world frame (green Gaussian) must respect both covariances and is calculated by covariance error propagation [19]:

$$\mathbf{C_y} = \mathbf{J_a}\mathbf{C_a}\mathbf{J_a}^T + \mathbf{J_w}\mathbf{C_w}\mathbf{J_w}^T \ , \tag{1}$$

where $\mathbf{C_y}$ denotes the covariance of the concatenated transformation $\mathbf{y} = \mathbf{g}(\mathbf{w}, \mathbf{a}) = \mathbf{w} \cdot \mathbf{a}$, and $\mathbf{C_a}$, $\mathbf{C_w}$ denote the covariance of $\mathbf{a}$ and $\mathbf{w}$, respectively. The Jacobians are given by $\mathbf{J_a} = \partial\mathbf{y}/\partial\mathbf{a}$ and $\mathbf{J_w} = \partial\mathbf{y}/\partial\mathbf{w}$. For clarity Fig. 1 visualizes the 2D case, while we normally use 3D transformations. We use the transformation framework of the MIRA middleware [20] which transparently handles the uncertainty propagation. Finally, the error-propagated Gaussians are aligned to the head position of people. The mean of each Gaussian is moved along the vertical axis to the expected head position. Furthermore, the vertical axis of the covariance is increased according to the uncertainty of the head position to the detected body part, e.g. high additional variance for leg detections accounting for different heights and poses of people, but none for face detections. In future work we want to learn the certainty of the sensor models and the parameters of the alignment from training data.

C. People Tracking

Our probabilistic people tracking system fuses Gaussians of multiple asynchronous detection modules. Figure 2 gives an overview of the people tracker and its processing steps described below.

1) Data Association: All Gaussian detections within the last 100 milliseconds are sorted by their detection time and processed sequentially. First, all hypotheses in the tracker are predicted up to the timestamp of the detection using the prediction step of the used filter algorithm (Sec. III-C.4). Second, the detection is assigned to the closest hypothesis in the tracker using the Mahalanobis distance:

$$d = (\mu_h - \mu_d)^T (\mathbf{C}_h + \mathbf{C}_d)^{-1} (\mu_h - \mu_d) \ , \tag{2}$$

where μ_h, $\mathbf{C}_h$, μ_d, $\mathbf{C}_d$ are the mean and covariance of the hypothesis and detection positions, respectively. The hypothesis with the smallest distance d is considered as responsible for the observation, and the update step of the filter algorithm is applied to improve the estimated hypothesis. If all calculated distances exceed an empirically determined threshold $d_{max} = 1.5$, the detection is considered as a new track, and a new hypothesis with a new filter algorithm is inserted at the detection's mean position μ_d with covariance $\mathbf{C}_d$.

Besides the uncertainty given by the covariances of the Gaussians, we introduce an additional confidence value to each hypothesis which captures the precision of each detector. While a leg detection is much more precise in position estimation (lower covariance of the Gaussian) than a HOG detection, the probability of being a person might be lower because many objects like chairs and tables produce false positives. When a detection is successfully assigned to a hypothesis the confidence of the hypothesis is increased by:

$$c_h := c_h + (1 - c_h)c_d \ , \tag{3}$$

where c_h is the confidence of the hypothesis and c_d is the confidence of the detection. c_h and c_d are limited to $[0, 1]$. c_d has a big influence on c_h if c_h is small and a small influence if it is close to 1. Finally, c_h represents the confidence of a hypothesis of being a person. In doing so we can validate hypotheses by the observation of multiple cues. By limiting the maximum confidence each sensor cue can add to the overall confidence, multiple cues are required to validate a hypothesis. Hence, detections from a single cue might create new tracks but are not outputted until a detection from another sensors is assigned to the track.

2) Covariance Intersection: Occasionally, a sensor input produces multiple detections on similar positions that would be fused in the data association step by the tracker. Examples are multiple bounding boxes of a visual detector that does not apply non-maximum suppression or overlapping image motion detections. Assuming that those detections originated from the same source, correlation between them is usually unknown. In that case, a Bayesian filtering algorithm, e.g. a Kalman filter, would underestimate the covariance of the detection by fusing all detections on the nearest hypothesis, because it assumes independence of the measurements.

Therefore, we apply covariance intersection [21] to fuse those detections to a single Gaussian:

$$\mathbf{C}_3^{-1} = (1 - \omega)\mathbf{C}_1^{-1} + \omega\mathbf{C}_2^{-1} \ , \tag{4}$$

where ω is a weighting parameter that defines the influence of the source covariances $\mathbf{C}_1$ and $\mathbf{C}_2$ on the resulting covariance $\mathbf{C}_3$. It is set to:

$$\omega = \frac{|\mathbf{C}_1|}{|\mathbf{C}_1| + |\mathbf{C}_2|} \ , \tag{5}$$

which balances the influence of both covariances [21]. The mean of the fused detection is calculated by:

$$\mu_3 = \mathbf{C}_3 \left[(1-\omega)\mathbf{C}_1^{-1}\mu_1 + \omega\mathbf{C}_2^{-1}\mu_2\right] \ , \qquad (6)$$

respecting the covariances of the considered detections.

3) Out-of-Sequence-Measurements: Although the tracker is sorting all detections in a time interval based on their timestamp, occasionally the current observation might be older than the current tracker state. The reason for that is the different processing time of the asynchronous detection modules. A connected laser leg detector produces frequent observations, while a HOG detector needs more time for processing one image. If processing of the observations is triggered while the HOG module still processes its image, the detection of the HOG is outdated in the next processing cycle of the tracker, because its timestamp (set by the processed image) is older than the timestamp of the current hypotheses in the tracker set by recent leg detections. To handle this out-of-sequence measurement (OOSM), the motion model of the tracker is skipped and the observation is predicted to the current timestamp using the predict method of its assigned filtering algorithm (Sec. III-C.4). The observation is then normally used to update the hypotheses in the tracker. A detailed analysis of the benefits of the OOSM modeling will be subject of future work.

4) Filtering: Generally, we designed the people tracker as a framework and allow for any filtering algorithm that can use Gaussian distributions as input and reflect its state as a Gaussian. As in our former work [22], we apply a 6D Kalman filter tracker that tracks the position and velocity of each hypothesis in the system [23]. The state space of a hypothesis is given by:

$$\mathbf{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z})^T \ , \qquad (7)$$

where x, y, z denote the 3D position and $\dot{x}, \dot{y}, \dot{z}$ the 3-dimensional velocity. Each hypothesis undergoes a normally distributed constant acceleration over the time interval $[\mathbf{x}_{k-1}, \mathbf{x}_k]$. Additionally, the confidence c_h of each hypothesis is lowered by a fixed time dependent value in the prediction step of the filter.

5) Hypotheses Management: The system comprises several mechanism to manage and limit the number of hypotheses. First, the tracker merges hypotheses with similar positions and velocities. Second, it prunes weak hypotheses with high positional covariance and low confidence, i.e. those that are not observed anymore. Third, detections and hypotheses in walls or obstacles can be pruned by using knowledge of the operation area, e.g. from an occupancy map which is also used by the robot for localization.

IV. EXPERIMENTS

We captured eight different data sets on our mobile platform [1]. The data sets are given in form of MIRA tapes [20] and contain rectified RGB images of the fish-eye front camera, LRF data, 3d range data of the Kinect sensor (Tab. I), intrinsic and extrinsic parameters of the cameras, coordinates of the different sensor frames, an occupancy map, odometry, and the robot's pose given by MCL. Note that our tracking system does not make use of the Kinect data so far. The data sets increase in difficulty (see Tab. II and Fig. 3). All people in the data set are manually labeled with bounding boxes in the RGB image, IDs, and occlusion information using the VATIC label tool [24]. The full data sets, pure jpg images, and label information are publicly available[1].

TABLE I

STATISTICS OF THE SENSORS

Sensor data	Format	Frequency
RGB images (rect. fish-eye)	800x600 px	15 Hz
Kinect Depth	640x480 px	10 Hz
LRF	Range vector	12 Hz
Robot pose	2D PoseCov	15 Hz

TABLE II

STATISTICS OF THE DATA SETS

Data set	Length	Frames	Info
Hallway	46 s	629	1-4 people walking
Follow	110 s	1679	following 1 person
Chair+Couch	82 s	1089	1 person sitting down
Sitting 1-4	218 s	2916	1-2 people sitting

We evaluated our real-time tracking system on the aforementioned data sets and compared it to offline trackers using state-of-the-art detection modules. The 3D Gaussians of the trackers are transformed into bounding boxes in the image. The height of each bounding box is calculated using the height of the corresponding Gaussian (top position) and assuming that people touch the ground (bottom position). The width of the transformed bounding box is determined empirically to half the size of the height. The bounding boxes and their IDs are compared to the labeled bounding boxes using the Multiple Object Tracking Performance (MOT) metric [25] which evaluates the precision, accuracy, and ID switches of the trackers. The intersection over union metric is used as a distance measure with a somewhat less restrictive threshold of 0.25 compared to the standard value of 0.5. The reason for this is, that we do not explicitly estimate people's poses but transform 3D Gaussians to bounding boxes in the image assuming a fixed height/width ratio. Hence, in case of sitting postures and almost quadratic labeled boxes, the overlap of the tracker's bounding box significantly reduces.

For each data set, we present the precision, recall and MOT metrics. The following tables show the mean misses ($\overline{\text{Miss}}$), the average false positives ($\overline{\text{FP}}$), the mean mismatch error ($\overline{\text{MME}}$), recall (RC), precision (PR), the multi object tracking precision (MOTP) and accuracy (MOTA) [25]. The first 3 values denote a ratio of accumulated misses, false positives, and mismatches over the total number of ground truth objects in the data sets, respectively. The MOTP denotes the average error in the estimated position for all matched hypothesis-label pairs. The distance of a match is calculated using intersection-union metric. Hence, the MOTP is bounded to the interval $[0, 1]$ with 0 being perfect and 1 being worst (no overlap of bounding boxes). Finally, the accuracy and

[1] http://www.tu-ilmenau.de/neurob/team/dipl-inf-michael-volkhardt/

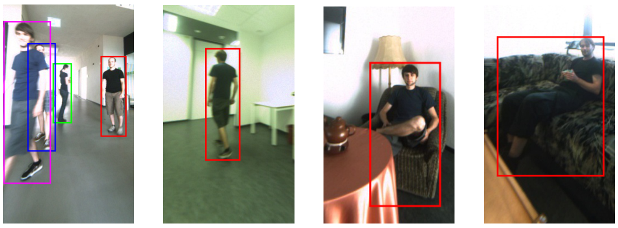
(a) Hallway (b) Follow (c) Chair+Couch (d) Sitting 1-4

Fig. 3. Exemplary labeled pictures of the different data sets. (a) Standing robot with multiple moving people, (b) robot following a person with another person passing by, (c) standing robot with person sitting down and standing up, (d) searching robot, person sitting and occasionally standing up.

TABLE III

RESULTS OF REAL-TIME AND LASER ONLY TRACKER

Data set	$\overline{\text{Miss}}$	$\overline{\text{FP}}$	$\overline{\text{MME}}$	RC	PR	MOTP	MOTA
Hallway	0.30	0.28	0.0109	0.76	0.73	0.50	0.40
- Laser	0.40	0.24	0.0100	0.66	0.73	0.51	0.35
Follow	0.26	0.28	0.0122	0.77	0.73	0.51	0.45
- Laser	0.24	0.39	0.0071	0.79	0.67	0.52	0.35
C.+C.	0.43	0.55	0.0066	0.59	0.51	0.54	0.02
- Laser	0.49	0.19	0.0102	0.52	0.74	0.53	0.32
Sit. 1-4	0.51	0.48	0.0044	0.49	0.52	0.61	0.01
- Laser	0.55	0.87	0.0094	0.45	0.44	0.63	-0.43

consistency of the tracker is given by the MOTA value:

$$MOTA = 1 - \frac{\sum_k (Miss_k + FP_k + MME_k)}{\sum_k G_k}, \quad (8)$$

where $Miss_k$, FP_k, and MME_k are the misses, false positives, and mismatches for time k, respectively and G_k denotes the number of all labels for time k. Here, a value of 1 means perfect tracking with no missed objects, no false positives and no identity switches. Note that the lower value of the MOTA is unbounded and can easily become negative - especially if there are false positives in the tracks.

The results of our real-time tracker, using face, HOG, upper-body HOG, motion and leg detections and a purely leg detection based tracker are given in Tab. III. Results of an offline FPDW tracker and a combined FPDW+leg detections based tracker are given in Tab. IV, while the results of the offline partHOG tracker and partHOG+leg detections based tracker are given in Tab. V. Furthermore, we give precision and recall values of the pure detectors in Tab. VI.

The real-time tracker shows good performance when people stand or walk, but performance quickly degenerates when people sit (Tab. III). Yet, it is superior to a purely leg-detection based tracker, except for the Chair+Couch data set where it produced a higher $\overline{\text{FP}}$ caused by consistent false positive HOG detection on a floor lamp. Overall the combination of multi-modal modules increases the tracking performance resulting in higher RC and PR values. The data sets where people sit reveal the limits of our tracking system. The system often misses people sitting calmly when there are no face or upper body detections (high miss rate for sitting scenarios). The sitting data sets also include more false positives mostly caused by the legs of cupboards and tables and person similar objects like a floor lamp, plants and a lamp on a cupboard.

TABLE IV

RESULTS OF FPDW AND FPDW+LASER TRACKER

Data set	$\overline{\text{Miss}}$	$\overline{\text{FP}}$	$\overline{\text{MME}}$	RC	PR	MOTP	MOTA
Hallway	0.51	0.34	0.0075	0.50	0.59	0.55	0.14
+ Laser	0.28	0.40	0.0174	0.77	0.66	0.56	0.29
Follow	0.40	0.22	0.0032	0.60	0.73	0.51	0.37
+ Laser	0.19	0.31	0.0032	0.82	0.72	0.53	0.48
C.+C.	0.835	0.44	0.0065	0.17	0.27	0.64	-0.28
+ Laser	0.66	0.45	0.0093	0.35	0.43	0.57	-0.11
Sit. 1-4	0.94	0.40	0.0033	0.06	0.10	0.72	-0.34
+ Laser	0.67	0.37	0.0058	0.33	0.54	0.55	-0.04

TABLE V

RESULTS OF PARTHOG AND PARTHOG+LASER TRACKER

Data set	$\overline{\text{Miss}}$	$\overline{\text{FP}}$	$\overline{\text{MME}}$	RC	PR	MOTP	MOTA
Hallway	0.42	0.16	0.0100	0.60	0.79	0.49	0.41
+ Laser	0.30	0.31	0.0125	0.74	0.70	0.49	0.37
Follow	0.28	0.16	0.0045	0.73	0.82	0.48	0.56
+ Laser	0.11	0.35	0.0045	0.95	0.73	0.49	0.54
C.+C.	0.46	0.44	0.0047	0.55	0.56	0.56	0.10
+ Laser	0.36	0.51	0.0093	0.65	0.56	0.56	0.14
Sit. 1-4	0.52	0.36	0.0032	0.49	0.54	0.60	0.12
+ Laser	0.39	0.39	0.0033	0.61	0.62	0.57	0.22

TABLE VI

RECALL AND PRECISION OF DETECTORS (OFFLINE ON EACH FRAME)

(a) FPDW

Data set	RC	PR
Hallway	0.76	0.97
Follow	0.53	0.98
Chair+Couch	0.21	0.81
Sitting 1-4	0.13	0.37

(b) PartHOG

Data set	RC	PR
Hallway	0.58	0.50
Follow	0.62	0.86
Chair+Couch	0.58	0.75
Sitting 1-4	0.53	0.65

The offline FPDW based tracker shows relative good performance for up-right pose people (Tab. IV). When people sit performance heavily decreases, which is due to the fact that the FPDW was trained for pedestrian detection. Yet, in all cases the performance can be increased when using an additional leg detector which helps to fill the gap of missing detections. Because our tracker also includes motion, face and upper-body detectors, its performance is superior to the FPDW and FPDW+laser based tracker - especially when people sit. On the other hand, a real-time CPU C++ implementation of the FPDW method would definitely improve our tracker when people are in an up-right pose.

Best results are achieved when using the offline partHOG based tracker (Tab. V). The high recall and precision values of the detector result in the highest MOTA values in almost all data sets. When combining the tracker with a leg detector, the leg detections help to fill missing detections resulting in a higher recall. On the other hand precision and the MOTA go down, because of many false positives of the leg detector.

The pure FPDW detector (Tab. VI(a)) often achieves better results than the FPDW based tracker. Reasons are that the tracker keeps hypotheses too long, data association distance and the motion model are a little too restricted for this set-up, and finally the projection of the 3D Gaussians

TABLE VII
PROCESSING TIME OF MODULES

Module	Avg. processing time [ms]	
	800x600 px	640x480 px
Face detector	172.4	99.7
Upper-body / HOG detector	408.8/423.0	242.3/225.4
Motion / Leg detector	3.1/1.0	1.6/1.0
FPDW (offline)	535.4	359.0
PartHOG (offline)	4975.7	2864.7
People Tracker	0.2	0.2

to bounding boxes is error prone, especially in distance estimation, because the camera is looking horizontally. These reasons need further investigations in future work. On the other hand, the combination of FPDW+laser and our real-time tracker achieve higher performances than the single FPDW detector, especially when people sit. The partHOG detector (Tab. VI(b)) achieved similar performance as the partHOG tracker, because the detector processed every frame in an offline evaluation.

All presented results were produced using the same set of parameter of the tracker and the detection modules. We scaled down the original image resolution of the data sets to 640x480 to increase the computational performance. A performance evaluation of the detection modules of the people tracker can be found in Tab. VII. From there it becomes obvious that the face and HOG modules do not process every frame but are set to run every 500 ms. The complete tracking system runs in real-time and is configured to consume 60% of the robot's on-board CPU (Intel i7-620M quad core processor) leaving enough space for the other required modules of the robot [1].

V. CONCLUSION

We presented a real-time, multi-modal people tracking system for mobile companion robots, that tracks walking people and is able to track people in sitting poses, if there are enough detector inputs. The system is evaluated on different data sets with increasing difficulty. Furthermore, we compared the performance to offline state-of-the-art people detectors like FPDW and partHOG and trackers based on these detectors. Our real-time version of the people tracker achieves better results than a tracker based on the FPDW detector and the pure detector, particularly when people sit. Best results are achieved when using the partHOG detector, which, unfortunately, is far from being real-time capable at the moment. Yet, the only moderate performances of all tested trackers show that more research is necessary to track people in home environments - especially for non-upright poses. To achieve the long-term goal of autonomous companion robots that support the elderly, we need to enhance current person detection algorithms. The modules for face and upper body detection are not robust enough to detect people in sitting postures or given occlusion. Using the real-time capable FPDW detector in the combined tracker could help to raise up-right posture performance. Real-time implementations of part based detection concepts like partHOG or poselets [7] that handle occlusion and multiple postures would greatly improve detection and tracking performance. Therefore, a major challenge lies in the development of real-time capable methods for detection of people in different poses, like sitting and lying. The Kinect sensor could help to achieve this goal.

REFERENCES

[1] H.-M. Gross et al., "Further progress towards a home robot companion for people with mild cognitive impairment," in *IEEE Trans. Syst., Man, Cybern.* Seoul, South Korea: IEEE, 2012, pp. 637–644.

[2] C. Pantofaru, "The Moving People, Moving Platform Dataset," http://bags.willowgarage.com/downloads/people_dataset/.

[3] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: an evaluation of the state of the art." *Transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[4] P. Dollár, S. Belongie, and P. Perona, "The Fastest Pedestrian Detector in the West," in *British Machine Vision Conference*, 2010.

[5] R. Benenson and M. Mathias, "Pedestrian detection at 100 frames per second," *Computer Vision and Pattern Recognition*, 2012.

[6] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2241–2248.

[7] L. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting people using mutually consistent poselet activations," in *ECMR*, 2010, pp. 168–181.

[8] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2002.

[9] N. Bellotto and H. Hu, "A Bank of Unscented Kalman Filters for Multimodal Human Perception with Mobile Service Robots," *International Journal of Social Robotics*, vol. 2, no. 2, pp. 121–136, 2010.

[10] G. Cielniak, T. Duckett, and A. J. Lilienthal, "Data association and occlusion handling for vision-based people tracking by mobile robots," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 435–443, 2010.

[11] K. O. Arras, O. M. Mozos, and W. Burgard, "Using Boosted Features for the Detection of People in 2D Range Data," in *International Conference on Robotics and Automation*, 2007, pp. 3402–3407.

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

[13] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled object detection and tracking from static cameras and moving vehicles." *TPAMI*, vol. 30, no. 10, pp. 1683–98, 2008.

[14] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *TPAMI*, vol. 33, no. 9, pp. 1820–1833, 2011.

[15] D. Mitzel, P. Sudowe, and B. Leibe, "Real-Time Multi-Person Tracking with Time-Constrained Detection," in *Proceedings of the British Machine Vision Conference*, 2011, pp. 104.1–104.11.

[16] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[17] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *CVPR*, 2008, pp. 1–8.

[18] M. Everingham et al., "The PASCAL Visual Object Classes Challenge 2009 Results," http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

[19] A. L. Simpson et al., "Uncertainty propagation and analysis of image-guided surgery," in *SPIE Medical Imaging: Visualization, Image-Guided Procedures, and Modeling Conference*, vol. 7964, 2011.

[20] E. Einhorn, T. Langner, R. Stricker, C. Martin, and H.-M. Gross, "Mira - middleware for robotic applications," in *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2591–2598.

[21] L. Chen, P. Armabel, and R. Mehra, "Estimation Under Unknown Correlation: Covariance Intersection Revisited," *IEEE Transactions on Automatic Control*, vol. 47, pp. 1879–1882, 2002.

[22] M. Volkhardt, S. Müller, C. Schröter, and H.-M. Gross, "Playing Hide and Seek with a Mobile Companion Robot," in *Proc. 11th IEEE-RAS Int. Conf. on Humanoid Robots*, 2011, pp. 40–46.

[23] D. F. Sebastian Thrun, Wolfram Burgard, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The Mit Press, 2005.

[24] D. R. Carl Vondrick, Donald Patterson, "Efficiently scaling up crowd-sourced video annotation," in *IJCV*, 2012.

[25] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," in *EURASIP Journal on Image and Video Processing*, 2008, pp. 1–10.