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

Finding People in Home Environments with a Mobile Robot

Michael Volkhardt and Horst-Michael Gross[1]

***Abstract*— Mobile robots assisting people in their apartment are subject to recent research efforts. To provide useful services a robot must be aware of the position of the user. Since the sensors of a robot have a limited field-of-view, the user is easily lost once he or she leaves the robot's vicinity. This paper proposes a method to search and locate a person that is distant – probably in another room – by driving through the apartment and checking for people. We apply restrictive as well as computationally expensive detection methods on-demand to verify the hypotheses of a real-time person tracker. Additionally, the search-tour of the robot is guided by an occurrence probability histogram of previous positions of the user. The system has been tested by evaluating accuracy and time the robot needs to find a user in a 3-room scenario.**

I. INTRODUCTION

The development of mobile companion robots assisting users in domestic environments is a long-term goal of recent research efforts [1]. These robots can support, entertain, or help elderly people to live independently for as long as possible in their homes. Supporting users in their daily routine and increasing their quality of life with intelligent tools 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 [2]. Since almost all services are built around human-machine-interaction, the robot needs to be aware of the user's presence and position in the apartment. Commonly person tracking systems are applied to detect and track people in the proximity of the robot. However, in a daily scenario, the user can easily leave the field-of-view of the robot, e.g. by going to another room. When user interaction is required, e.g. if a video-call comes in or a reminder has to be delivered, the robot needs to drive through the apartment and look for the user.

On its search tour, the robot should locate the user quickly and accurately. A system that needs several minutes to deliver an incoming video call or tries to interact with false positive hypothesis is not acceptable. Therefore, we improved the robustness of our person tracking system by verifying hypotheses on demand with modules that cannot be run in parallel all the time. These include a motion module, that can only be applied when the robot is standing still, and a computationally expensive partHOG detection module [3], which needs several seconds to process one image. Second, we mark hypotheses that do not lead to a user interaction as false positives and ignore them subsequently. Finally, we developed a method to guide the search tour of the robot by checking positions in the apartment with a high user occurrence probability based on previous observations. Because we aim for a practical solution, all methods run in real time on the robot's hardware. Note that we assume that there is only one person in the apartment. Generally, the method is not restricted to a single user. Since we do not use any person recognition, the robot would then just find and stop at the first user in sight. The remainder of this paper is organized as follows: Section II summarizes related work in the research area. Sec. III briefly presents our person tracking system and Sec. IV addresses the search behavior of the robot. Afterwards, Sec. V gives a description of experimental results and Sec. VI summarizes our contribution and gives an outlook on future work.

II. RELATED WORK

People detection and tracking are well-covered research areas, and impressive results have been accomplished in recent years. A huge amount of visual detection methods originated from the field of pedestrian detection, each with their own benefits and disadvantages (see [4] for a survey). Recent approaches like [5], [6] achieve good results at frame-rate by applying a soft-cascade, tuning features, sampling the image pyramid, and by using ground plane constraints. Note that pedestrian detection only covers a part of the problem of finding people in their homes, e.g. related to the variety of poses encountered. In contrast, [3] achieves impressive results, given partial occlusion and varying poses, but requires several seconds per image. In the field of mobile service robotics, person detection in camera images mainly focuses on the face [7] or the aforementioned pedestrian detection methods. Additionally, most mobile robots are equipped with a laser range finder (LRF) that allows the detection of human legs [8].

Plenty of research has been done to develop methods for people tracking on mobile robots in real-world applications. Often, evaluation is done on pre-captured data and real-time performance retreats into the background while the main focus concentrates on tracking quality [9], [10], [11]. The method of [12] tracks people in real-time but requires special hardware, like a stereo camera and a dedicated GPU which are both not available on our mobile platform [1]. Our practical solution uses well established methods like HOG, motion, face, and leg detection, whose detection quality is moderate compared to aforementioned cutting-edge methods. This is mainly because our robot has limited

[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

processing power, which must be shared by other modules, e.g. for navigation and interaction. Promising new detection paradigms could not be applied in this work, because they are only implemented in Matlab so far [5]. Note that we cannot apply methods that depend on GPU [6] offline because the robot directly reacts on the output of the people tracking system.

To the best of our knowledge, there is very little work on searching and finding people in home environments. In a former work, we developed a method to search people by checking predefined places where people usually rest, e.g. chairs and sofas [13]. The appearance of an empty place was learned using a multi-modal color model that captured daytime and viewpoint. Furthermore, a similar color model of the user's clothes was learned. By comparing the current impression of the place's appearance to the model of the empty place and the user model, a SVM decided if the place was occupied by the user or not. The system was accompanied by a person tracker that helped to detect standing and frontal sitting people. The search tour was guided by external feedback of infrared motion sensors installed in the apartment that gave a coarse location of the user. The method has two main drawbacks: First the places had to be learned in different illumination conditions and daytimes beforehand. This training required that the user was not at home. Second, the learned color model of the user and the place had to differ. In this work, we follow a different approach and try to enhance our person tracker by verifying hypotheses, lowering false positives, and developing an intelligent search strategy. This search strategy selects different points in the apartment to drive to. We apply a greedy strategy, which selects the navigation point that is closest to the robot or has the highest probability of the user's presence. The field of perception planning offers more advanced exploration strategies to cover the whole apartment that could be integrated in the future [14], [15], [16]. We do not rely on infrared sensor hints as they are not available in every apartment and often require time-consuming installations.

III. PERSON TRACKING SYSTEM

Our probabilistic person tracking system fuses the output of multiple detection modules. To detect people, we make use of a persons' legs, face, motion, and body-shape. The leg detection module applies a boosted cascade of classifiers in the range data of a laser range finder to classify segments as legs [8]. Afterwards, pairs of legs are merged to positions of people. The face detection system utilizes the well-known AdaBoost detector of Viola&Jones [7]. Each time the robot does not move, which is signaled by the robot's odometry, we apply a simple image difference detection. After thresholding and a connected components algorithm, we get bounding boxes of moving regions in the image. Furthermore, we apply a full body and an upper body shape detector based on Histograms of Oriented Gradients [17], [18] with a ground plane constraint. Each module detects persons by different body parts, e.g. the face, legs, or head-shoulder contour. To facilitate fusion in the person tracker, we transform the detections to a world coordinate frame and align them to a common reference point, i.e. position of the head [13]. New position observations are transformed into Gaussians in 3D world representation and the positional uncertainty of the detection is respected in the covariance of the Gaussian. Figure 1 gives an overview of the person tracker and its processing steps which are described below.

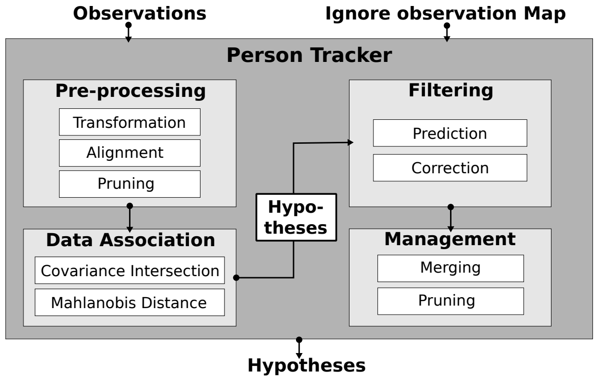

Fig. 1. Overview of the processing steps of the person tracker.

A. DATA ASSOCIATION

All detections within the last 100 milliseconds are sorted by detection time and are processed sequentially starting with the oldest one. First, all hypotheses in the tracker are predicted up to the timestamp of the observation using the prediction step of the used filter algorithm (Sec. III-B). Then, the observation is assigned to the closest hypothesis in the tracker if their Mahalanobis distance is below an empirically set distance $d_{max} = 1.5$:

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

where μ_h, $\mathbf{C}_h$, μ_d, $\mathbf{C}_d$ are the mean and covariance of the hypothesis and detection positions, respectively. Otherwise, a new hypothesis is created.

1) 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 motion detections. Assuming that those detections originated from the same source, correlation between them is usually unknown. In that case, a 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 measurement which does not hold in this case. Therefore, we apply covariance intersection [19] to fuse those detections to a single Gaussian:

$$\mathbf{C}_3^{-1} = (1-\omega)\mathbf{C}_1^{-1} + \omega\mathbf{C}_2^{-1} , \quad (2)$$

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:

$$\omega = \frac{|\mathbf{C}_1|}{|\mathbf{C}_1|+|\mathbf{C}_2|} , \quad (3)$$

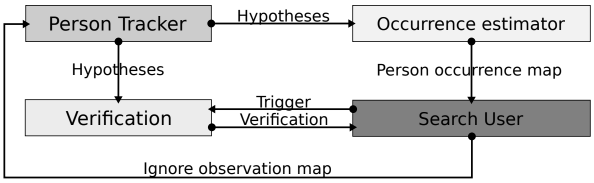

Fig. 2. Dependencies of modules involved in the search behavior.

which balances the influence of both covariances [19]. The mean of the fused detection is calculated by:

$$\mu_3 = \mathbf{C}_3 \left[(1-\omega)\mathbf{C}_1^{-1}\mu_1 + \omega\mathbf{C}_2^{-1}\mu_2 \right] \ , \qquad (4)$$

respecting the covariances of the concerned detections.

B. FILTERING

Generally, we designed the person tracker as a framework and allow any filtering algorithm that can use Gaussian distributions as input and reflect its state as a Gaussian. As in our former work [13], we apply a 6D Kalman filter tracker that tracks the position and velocity of each hypothesis in the system. The state space of a hypothesis is given by:

$$\mathbf{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z})^T \ , \qquad (5)$$

where x, y, z denote the 3D position and $\dot{x}, \dot{y}, \dot{z}$ the 3-dimensional velocity.

C. HYPOTHESES MANAGEMENT

The system comprises several mechanism to manage and limit the number of hypotheses. First, the tracker merges hypotheses with similar position and velocities. Second, it prunes weak hypotheses by removing those with high positional covariance, i.e. those that are not observed anymore. Third, hypotheses in walls or obstacles can be pruned by using an occupancy map which is also used by the robot for localization and path planning. Finally, the tracker can ignore detections and prune hypotheses that lie on marked areas of a so-called "ignore observation map" that is similar to an occupancy map. This mechanism can be used to mark areas where the privacy of the user should be respected, e.g. in the bathroom. Furthermore, it is also used by the search behavior to discard false positives.

IV. SEARCH BEHAVIOR

The previous section described how the robot can detect and track people when they are in the field-of-view of its sensors. In a typical home scenario, the user can quickly go to another room and, therefore, easily leave the robot's detection range. As it is not desired that the robot always closely follows the user, the robot needs a mechanism to search a person in the apartment. Our dialog application of the robot already makes use of a list of navigation points defined by the user to send the robot to a location in the apartment, e.g. near the couch or into the kitchen. Those navigation points include the position and orientation of the robot and are used to guide the search tour of the robot.

The general process and the dependencies between the modules involved in the search process are shown in Fig. 2. When the search behavior is triggered, the robot starts driving to the closest navigation point using adaptive Monte Carlo localization, E*-path planning and motion control based on the dynamic window approach [1]. Once the robot reaches a navigation point, it marks it as visited and chooses the next closest one from the list. The navigation points serve as a hint where to look for people. In Sec. IV-C we describe how to improve the selection scheme. While the robot drives to a navigation point, the search module checks for hypotheses from the person tracker. When one or more hypotheses are found, it stops and turns to the closest one and triggers a verification process (Sec. IV-A). This verification labels hypotheses as certain or uncertain. Usually the hypotheses include strong true positives, i.e. detected by multiple sensor cues, weak true positives, i.e. only observed by one cue and false positive, i.e. either from a single or multiple cues. If none of the hypotheses has been verified, the robot drives closer to the nearest hypothesis and restarts the verification process. If the hypothesis is still not verified, it is marked as a false positive (Sec. IV-B), and the robot continues its search tour. If a hypothesis has been verified, the robot starts approaching it, which means driving very closely while respecting the orientation towards it. Then the robot waits some time for any input through the touchscreen, which normally automatically happens if the user answers to the robot's interaction request. If an input is recognized the search behavior ends. Otherwise, the hypothesis is also marked as a false positive. This heuristics causes the robot to eventually reach the user even if the verification mechanism fails and verifies a false positive of the tracker.

A. HYPOTHESES VERIFICATION

The verification is triggered on demand each time the search behavior wants to verify a hypothesis. We apply two detection modules that cannot be run parallel to the person tracker all the time. First, we use a computationally expensive partHOG with a VOC2009 model [3], [20]. This module reaches high average precision given partial occlusion with the drawback of taking 4-5 seconds to process an image making use of a parallel CPU implementation. For this time the robot physically stops its tour and pauses some expensive modules to allow fast processing of the image. The partHOG module detects sitting and partially occluded people. Second, because the robot is standing still, we can apply a motion module similar to the one that is already used by the person tracker. In contrast, this module also detects small motions like people leaning towards the robot, turning their heads, or waving with one hand. We noticed that these are all natural movements of users to signal the robot to come closer.

Without this verification process, the robot would need to drive to and approach every uncertain hypothesis of the person tracker and wait for user input. Because the person tracker occasionally produces false positives, the verification process should significantly decrease the time it takes to find the user in the apartment.

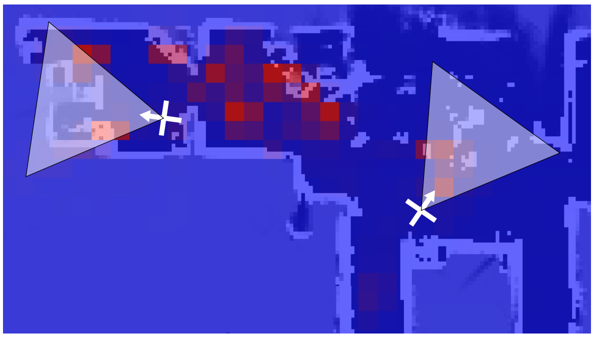

Fig. 3. Person occurrence map overlayed on the occupancy map. Probability of person occurrence from blue to red (scaled for clarification). White crosses - exemplary navigation points. Triangles - field of view of sensors used to integrate person occurrence probability.

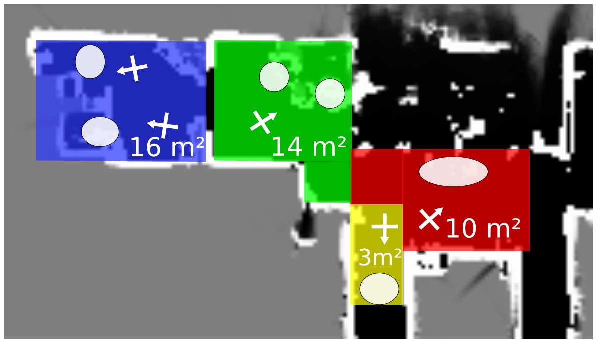

Fig. 4. Occupancy map of experimental environment with overlayed labels. Blue - living room, green - guest room, red - kitchen, yellow - hallway. White ellipses - location of user in experiments. White crosses with arrows - navigation points with orientation.

B. HANDLING FALSE POSITIVES

As stated before, hypotheses that are not verified by the aforementioned process or by user input are marked as false positives. We insert a rectangle region with a scenario dependent width (2 m) around the hypothesis into the "ignore observation map" (Fig. 2). Thus, the person tracker ignores detections and prunes hypotheses in this region (Sec. III). Hence, the robot can continue its search tour and look for people in other areas of the apartment. Without this mechanism the robot could endlessly wait in front of a, possibly even verified, false positive. To prevent marking too many areas in the apartment, we clear the regions after a certain time, e.g. one search tour. This allows the person tracker to detect people in those regions again.

C. PERSON OCCURRENCE MAP

The selection of navigation points in the search tour of the robot can be enhanced if we select those points close to positions that have a high probability of people's occurrences. Therefore, a module estimates a person occurrence probability map of the environment taking the hypotheses of the person tracker into account. This map is encoded as a 2D histogram, where the bins represent discretized rectangle areas of the apartment and each count in a bin signals a hypothesis in this area. The robot cannot simply drive to the bin with the highest probability because the destination might lie in a wall or the field of view would not correctly cover the area of the bin. Instead, the robot selects a navigation point at which its field-of-view covers an area with a high occurrence probability. Hence, we model the detection range of the robot as a triangle with one corner on the position of the navigation point and the perpendicular bisector of the triangle corresponding to the orientation of the navigation point (Fig. 3).

By integrating the area covered by the triangle in the person occurrence map, we can calculate the probability of a person's occurrence for each navigation point. The robot now selects the navigation point with the highest probability instead of the closest one. Once it has checked the area, the navigation point is labeled as visited and the second most likely point is selected. By using this mechanism, the robot first checks areas where the user was frequently detected in the past. We expect that this significantly lowers the time for finding the person, if the person still acts under the distribution of the learned model. Of course, if the user is in an unusual spot where he or she was rarely before, the search-tour duration could be worse.

V. EXPERIMENTAL RESULTS

The operational area of the experiments is a three-room scenario including a living and guest room, a kitchen, and a corridor. Figure 4 gives an overview of the arrangement of the rooms and their extensions. Furthermore, the navigation points are given as crosses with an arrow indicating the orientation of the robot. The location of the user in the experiments is visualized by circles. The user was sitting on two different locations on the couch in the living room, on two arm-chairs in the guest room, one chair in the kitchen that was moved around a table, and finally standing in the hallway. The exact position and orientation of the user on each location was slightly varied during the experiments. Although, our person tracker only robustly detects people in an upright position, with the enhancement presented above it is possible to robustly find the user in a sitting pose as well. This means that the tracker often produced an uncertain hypotheses of sitting people, i.e. through legs or a single HOG detection, and this hypothesis is verified subsequently. There is a balcony behind the kitchen in Fig. 4 which is separated by big windows. These caused frequent reflections and motions from objects outside (cars, plants, clouds).

A. VERIFICATION

First, we evaluated if our verification mechanism does improve the ability of the robot to find and locate the person in the apartment. These experiments did not use the person occurrence map to guide the search tour (Sec. IV-C). Yet, all experiments include the handling of false positives (Sec. IV-B). We divided the experiments into three parts to test different methods. The first method does not apply any verification mechanism (NoVer), the second method uses only motion verification (MoVer) and the third method uses the partHOG and motion verification (HoVer).

TABLE I

SUCCESS RATE AND DURATION OF SEARCH TRIALS FOR DIFFERENT METHODS. DURATION GIVEN FOR SUCCESSFUL TRIALS.

method	suc. rate	avg. t [s]	max t [s]	std t [s]
NoVer	0.72	53.1	146	43.9
MoVer	0.87	59.4	116	34.0
HoVer	0.91	55.5	115	34.6

To evaluate each method, we defined three starting locations for the robot to begin its search tour, i.e. in the living room, in the guest room, and in the kitchen. The user located itself on one of the six positions in the apartment (Fig. 4). The search tour to each location is repeated three times leading to $3 \times 6 \times 3 = 54$ trials for each method. We measured the success rate of the search behavior by counting a successful trial if the robot correctly found, approached, and stopped in front of the user, and the user was easily able to reach the touchscreen to end the search tour. We counted a failure if the robot did not find the user, did not approach the user, hit an obstacle, or if the search tour took more than 3 minutes. For comparison: the time the robots needs to drive to each navigation point starting from the position in the guest room without checking for any hypotheses is 80 seconds averaged over 5 trials.

The results of the trials are shown in Tab. I. The durations to locate the user are only calculated on the successful trials. The success rate of NoVer is significantly lower than MoVer and HoVer. HoVer achieved a higher success rate compared to MoVer, because the partHOG module detected sitting persons that did not move or whose legs where occluded. The most frequent reasons for failure in the trials (in order of occurrence) were:

1) the robot hitting an obstacle in the rooms,
2) the search tour took more than 3 minutes,
3) the robot completely missed the user,
4) the robot did not verify the user.

The problem of hitting obstacles is significantly increased for method NoVer because the robot does not verify hypotheses and approaches many false positive, e.g. behind windows, on tables, or cupboards. Although, the robot normally perceives these objects and avoids them, occasionally it bumpers when turning and navigating close to those obstacles. The second problem only appeared when using the NoVer method (see below). Missing the user occurred roughly equally likely with all methods. This sometimes happened if the user was in an unfavorable pose not sitting frontal to the robot's camera or with the legs occluded. Finally, when using the MoVer or HoVer method, the robot might not verify a hypothesis by motion or partHOG and ignores it as a false positive. Yet, the verification failed only two times in the experiments.

The average and maximal duration of the successful trials (Tab. I) are relatively large compared to the 80 seconds it takes the robot to visit all navigation points. When using NoVer, the duration is mainly increased by the robot approaching uncertain hypotheses and waiting for user input

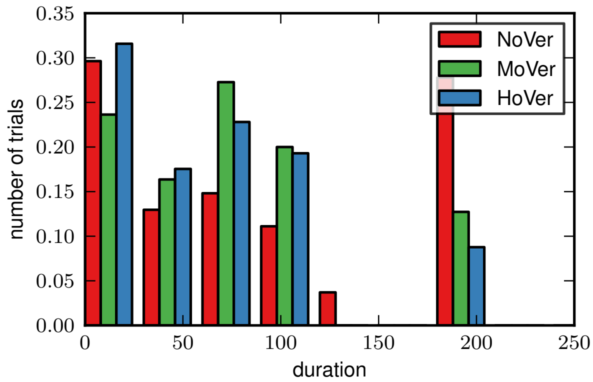

Fig. 5. Histogram of the search durations of the different methods

until finally marking this hypothesis as a false positive if no input is given. When using MoVer or HoVer the robot only turns to the hypotheses and verifies it. Yet, the robot needs time to apply the verification modules. This results in a slightly bigger average duration. This becomes most apparent in trials, where the position of the user is close to the robot's starting location. There are less uncertain hypotheses where the robot could save time by not approaching them. Additionally, the robot needs time to verify the hypothesis of the user whereas the NoVer methods just drives to it. On the other hand, the maximum time and the standard deviation of the verification methods is lower than when using the NoVer method.

For further clarification, the difference in the duration of all search trials is displayed as a histogram in Fig. 5. The width of the bin is set to 30 seconds, and we plot the duration counts for all three methods in the center of each bin. We set the search time of all failure trials, e.g. also those where the robot hit an obstacle, to 180 seconds which is equal to the maximum time we allowed the robot to find the user. Hence, the values in the last bin also include the unsuccessful trials. The NoVer method either is very fast (bin 1) or takes very long (bin 5 and bin 6) or completely fails (bin 6). The MoVer and HoVer method caused many mediocre duration trials, but also with many fast trials and few failures.

To summarize: Without the verification, the robot approaches each hypotheses. This could result in a lower search duration, but also lowers the success rate of finding the user in a reasonable time. By using the verification methods, the robot needs more time for verification, but yields a higher success rate. Therefore, we advocate the use of the verification mechanism.

B. *PERSON OCCURRENCE MAP*

To evaluate the usefulness of the person occurrence map (Sec. IV-C), we first estimated the map by tracking the user performing short sequences of daily activities and during the experiments of Sec. V-A. The resulting map looks similar to Fig. 3 with the highest probability of occurrence in the living room, followed by the guest room, kitchen, and hallway. We replace the navigation point selection of the HoVer method

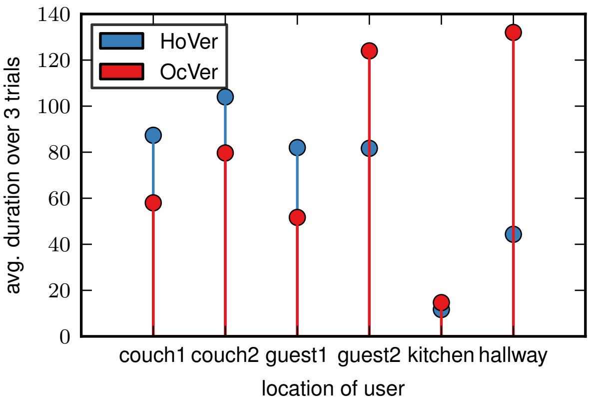

Fig. 6. Durations of successful search tours when using HoVer and CoVer methods.

by the occurrence map selection described in Sec. IV-C and call this method OcVer. To show the difference between the two methods, we set the starting location of the robot to the kitchen and repeated the experiment of Sec. V-A. We measured the duration it took the robot to reach the user on each of the six positions in the apartment averaged over three trials. The results are shown in Fig. 6. Although the robot is starting in the kitchen, the time it takes the robot to find the user on the couch1, couch2, and guest1 position are shorter when using the OcVer method. This is because the robot does not check the kitchen and hallway position and directly drives to the living room. While the robot drove to the living room, it also detected the user on the guest1 position resulting in a short search duration. A problem becomes apparent when the user is sitting on the guest2 position. The robot first drives through the guest room to the living room and does not detect the user on the arm-chair first, but only when checking the guest room later. This results in a much higher average search duration. This becomes extreme when the user is in the hallway. Although, physically close to the starting location, the robot ignores this room and checks the other navigation points first, resulting in a very high search duration. When the user was sitting in the kitchen, the robot still detected it when it started driving to the living room. In future work, we want to solve the problem of ignoring close navigation points by including advanced exploration strategies [14], [16] and combine them with the occurrence probability map.

VI. CONCLUSIONS

We presented a method to search and locate people in home environments by using a mobile robot. While driving through the apartment, the robot incorporates the detections of multiple person detection modules into a multi-modal person tracker. To enhance the search behavior, we use a motion and a partHOG detector to verify the hypotheses of the person tracker and prune false positives. Experiments of over 150 search trials substantiate that the implemented verification mechanism improves the success rate of finding the user and lowers the average search time. By modeling the usual occurrence probability of the user, the average search time can be further lowered if the user acts under the previously learned model.

In future work, we plan to enhance the person occurrence model by including contextual information like the time of the hypotheses, e.g. the user is usually in the kitchen at noon and on the couch in the evening. Another aspect is using more advanced exploration strategies based on the person occurrence probability and the map of the apartment [16]. Additionally, one could use other verification modules, like person detection in data of the kinect sensor or any kind of color and texture models.

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. Huijnen, A. Badii, and H. van den Heuvel, “Maybe it becomes a buddy, but do not call it a robot - seamless cooperation between companion robotics and smart homes,” in *Ambient Intelligence*, 2011, pp. 324–329.
[3] 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.
[4] 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.
[5] P. Dollár, S. Belongie, and P. Perona, “The Fastest Pedestrian Detector in the West,” in *British Machine Vision Conference*, 2010.
[6] R. Benenson and M. Mathias, “Pedestrian detection at 100 frames per second,” *Computer Vision and Pattern Recognition*, 2012.
[7] P. Viola and M. Jones, “Robust real-time object detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2002.
[8] 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.
[9] 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.
[10] A. Ess, B. Leibe, K. Schindler, and L. van Gool, “Moving obstacle detection in highly dynamic scenes,” in *International Conference on Robotics and Automation*, 2009, pp. 56–63.
[11] 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.
[12] 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.
[13] 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.
[14] I. M. Rekleitis, G. Dudek, and E. E. Milios, “Multi-robot exploration of an unknown environment, efficiently reducing the odometry error,” in *Proceedings of the 15th IJCAI - Volume 2*, 1997, pp. 1340–1345.
[15] J. Armingol, L. Moreno, A. De la Escalera, and M. Salichs, “Landmark perception planning for mobile robot localization,” in *IEEE Int. Conf. on Robotics and Automation*, vol. 4, 1998, pp. 3425–3430 vol.4.
[16] R. Sim and N. Roy, “Global a-optimal robot exploration in slam,” in *Proceedings of the 2005 IEEE ICRA*, 2005, pp. 661–666.
[17] 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.
[18] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, “Progressive search space reduction for human pose estimation,” in *Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
[19] 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.
[20] M. Everingham et al., “The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results,” http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.