Proc. 11th IEEE-RAS Int. Conf. on Humanoid Robots 2011 (Humanoids 2011), Bled, Slovenia}, pp. 40-46

Playing Hide and Seek with a Mobile Companion Robot

Michael Volkhardt, Steffen Mueller, Christof Schroeter, Horst-Michael Gross
Neuroinformatics and Cognitive Robotics Lab
Ilmenau University of Technology
98684 Ilmenau, Germany
Email: Michael.Volkhardt@tu-ilmenau.de

***Abstract*—This paper addresses the problem of finding people in domestic environments utilizing a mobile robot. Companion robots, which should provide different services, must be able to robustly estimate the user's position. While detecting people in an upright pose is mainly solved, most of the users' various poses in living environments are hard to detect. We present a tracking framework that incorporates state-of-the-art detection modules, but also a novel approach for visually detecting the presence of people resting at previously known seating places in arbitrary poses. The method is based on a contextual color model of the respective place in the environment and a color model of the user's appearance. The system has been tested by evaluating the robot's capability to find the user in a 3-room apartment in a hide and seek scenario.**

I. Introduction

This work is part of the CompanionAble[1] project, which aims to develop a personal robot for assisting elderly people with mild cognitive impairments. The goal of the project is to increase the social independence of users by means of a combination of a smart home and a mobile robot. Therefore, the system provides different services, like e.g. day-time management or video conferences with medical attendants, relatives, and friends. Furthermore, it recognizes emergency situations, like falls, and tries to prevent progression of the cognitive impairments by providing interactive stimulation programs. To offer these service functionalities, the robot system provides several autonomous behaviors. First, observing the user in a non-intrusive way allows to facilitate services that require interaction or to react on critical situations. A second behavior is following and approaching the user if interaction is desired. Third, the robot must seek for the user if a reminder has to be delivered or a video call comes in and the user is not in direct proximity of the robot. This work addresses the last mentioned behaviour.

A prerequisite to these behaviors is the robust detection and tracking of the user in the apartment. In contrast to other interaction applications in public environments, people in home environments often do not face the robot in an upright pose but sit on chairs or lie on sofas. Therefore, our system combines state-of-the-art methods for up-right pose people detection with a module to detect users independent of their pose at places, where they usually rest. The key idea of this module is to learn color-based models of the user's appearance and predefined resting places beforehand. In the detection phase, the current visual impression is compared to both of these models to decide if the learned user is present. Given a seeking task, the robot system is optionally supported by infrared motion sensors of the smart home system. These sensors detect motion in the apartment and can be used as a hint where to search first. Evaluation of the approach has been done by playing games of hide and seek. This means that the user "hid" somewhere in the apartment and the robot had to find her or him starting from a fixed location. Success rate and search time were used as performance measures, because reminders and incoming video calls should be delivered fast and accurate by the robot. Therefore, the contribution of this work is two-fold: First, we present a novel method for mobile robots that goes beyond the state-of-the-art by detecting people in situations not captured by common detection and tracking systems in living environments. Secondly, the performance of the tracking framework and its modules is extensively evaluated in multiple hide and seek runs. These experiments are to assess the reliability of the autonomous mobile companion robot under real-world conditions.

The remainder of this paper is organized as follows: Section II summarizes previous work carried out on the research topic. Sec. III presents the tracking framework. Sec. IV addresses the innovation of detecting lounging people at places in detail. The subsequent section describes the integration of infrared motion sensors to enhance the robot's search behavior. Afterwards, Sect. VI gives a description of the experiments carried out, while Sec. VII summarizes our contribution and gives an outlook on future work.

II. Related Work

People detection and tracking are prominent and well-covered research areas, and impressive results have been accomplished in recent years. Considering the constrained hardware of mobile robots, two main fields for people detection have been established – range-finder-based and visual approaches. Arras et al. [1] employ AdaBoost on laser range data to combine multiple weak classifiers to a final strong classifier that distinguishes human legs from the environment. Visual approaches mainly focus on the face or the human body

This work has received funding from the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216487.

[1] www.companionable.net

shape. The most prominent up-to-date face detection method was presented by Viola&Jones [2]. It utilizes AdaBoost to learn a cascade of simple, but very efficient image region classifiers to detect faces in camera images. Histograms of Oriented Gradients (HOG) [3] have been established as the state-of-the-art method for upright people detection. The basic idea is to compute block-wise histograms of gradient orientations, resulting in robustness to slight spatial variation of object shape, color, and image contrast. The histograms inside a detection window are concatenated into a high-dimensional feature vector and classified by a linear Support Vector Machine. Further extensions to the original HOG method focus on upper body detection [4] or use deformable sub-parts, which increase detection performance given partial occlusion [5]. Detection, segmentation and pose estimation of people in images is addressed by Bourdev et al. [6] who combine HOG features with the voting scheme of the Implicit Shape Model [7]. Schwartz et al. [8] augment the HOG features with color and texture information achieving impressive results on outdoor data sets. Unfortunately, the latter two approaches are far beyond real-time capabilities.

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 and single poses [3], [7], [9]. Yet, few approaches handle the detection and tracking of people in home environments, especially on mobile robots [10], [11]. Often smart home technologies like static cameras with background subtraction methods [12] or infrared sensors [13], [14] are applied, which facilitate the problem of detection. In conjunction with these findings, we rely on infrared motion sensors to support the tracking system of the mobile robot. On occasion, approaches working with mobile robots process the data captured offline to apply computationally heavy detection methods [9]. The CompanionAble project aims to develop a mobile robot companion that is able to react on and interact with the user during movement. Therefore, all those approaches employing background subtraction or a retro-perspective analysis are not applicable.

III. User Detection and Tracking

Typical scenarios in a home environment include the user walking to another room, or the user sitting on a chair or lying on a sofa. In the first case we are interested in tracing the user's trajectory to follow her or him or have a clue where to search first when an event calls for that. The detection of the user in the latter case is described in the next section.

Our tracking system comprises a multi-modal, multi-cue tracking framework based on the Kalman Filter update regime similar to an earlier approach of us [11]. The advanced system handles a set of independent 3D position hypotheses of people, which are modeled by Gaussian probability distributions. Adding the velocity results in a six dimensional state space $\mathbf{s} = (x, y, z, v_x, v_y, v_z)$ for each hypothesis. We use the head of the user as the reference for alignment. Therefore, z denotes the height of the user's head. The tracking system is designed in a framework-like fashion to incorporate the detections of arbitrary observation modules. New (asynchronous) position observations are transformed to the 3D representation of the tracking system. When using range-based detection modules, a Gaussian is created at the x, y position of the range measurement with a height value set to the common size of a person $z = 1.70$. In case of visual detection modules, we transform the bounding box of the user into a 3D Gaussian by using the parameters of the calibrated camera and estimating the distance through the size of the bounding box. The detection quality of the respective sensor is incorporated into the covariance of the Gaussian distribution, i.e. laser-based detection results in low variance in distance and direction but in large variance in height while visual detections have a high variance in distance estimation. Each resulting detection is associated with the closest hypothesis in the system. If a distance threshold is exceeded, a new hypothesis is introduced. Once the system knows the associated source for that observation, the position of that hypothesis is updated using the Kalman filter technique.

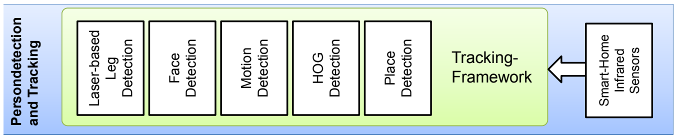
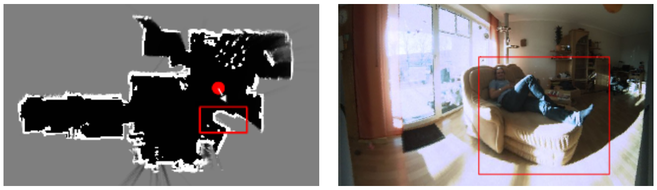

Persondetection and Tracking
Laser-based Leg Detection
Face Detection
Motion Detection
HOG Detection
Place Detection
Tracking-Framework
Smart-Home Infrared Sensors

Fig. 1. Overview of the tracking framework. In this work, we used 4 different observation cues to feed the tracker. The innovation is a module to detect people lounging at places. Furthermore, optional infrared motion sensors can guide the behavior of the robot.

In this work, we apply laser-based leg detection and multiple visual detection modules (Fig. 1). The first module is based on the boosted classifier approach of [1] and discovers legs in laser-range data. By searching for paired legs, the system produces hypotheses of the user's position. The face detection system utilizes the well-known face detector of Viola&Jones [2]. The motion detection cue [10] is only active when the robot is standing still and utilizes a fast and simple image difference approach. Furthermore, we apply a combination of a full-body HOG detector [3] and an upper body detector [4]. The system described so-far is able to detect and track upright standing people (mainly through legs and HOG) and people sitting in frontal-view (mainly through face and upper-body HOG) in the surroundings of the robot. In the following, we present a training-based method for detecting people in more difficult poses.

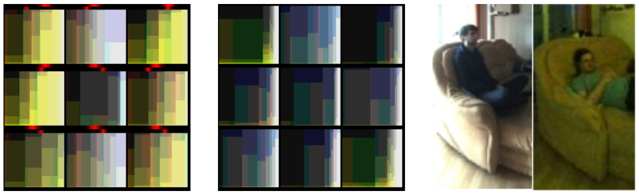
IV. Detection of Lounging People at Places

Lounging people are encountered quite frequently in a domestic environment, e.g. when the user is watching TV, reading newspaper, making phone calls, working or sleeping. Therefore, we developed a method that first learns the appearance of places in the apartment where the user usually rests. Afterwards, the deviation of occupied places from the respective models and the similarity to an up to date user model are used for detection.



(a) Place in occupancy map (b) Projection to camera image

Fig. 2. Place definition. (a) Bounding box of place in the occupancy map. The robot is in its observation position (red circle). (b) Place's bounding box projected into the camera image.



(a) Place model (b) User color model (c) Place examples

Fig. 3. (a) Place model with 9 color histograms and context distributions. (b) User color model with 9 color histograms. (c) Example of place given different illumination conditions and the user wearing different clothes.

A. *Definition of Places*

We define places as positions in the apartment where the user is usually encountered, e.g. chairs, sofas, working desk. Each place P is represented by a 3D box $\mathbf{b} = (x, y, z, d_x, d_y, d_z)$ with x, y, z being the center coordinates of the box and d_x, d_y, d_z denoting the width in each dimension. Figure 2 shows an exemplary place position in the world centered occupancy map used for navigation and the 2D-projection of the place box into the current camera image of the robot. The projection is done by using the robots current position in the aparment given by Monte Carlo localization [15] and the projection matrix of the calibrated camera. Naturally, the content of the place-boxes looks completely different in the camera image, if observed from different positions. Since the system is learning the appearance of a number of places in the apartment, we need to restrict the pose from which the robot is observing them. Therefore, each place is assigned n observation poses $\mathbf{O} = (\mathbf{o}_1, \ldots, \mathbf{o}_n)$, where $\mathbf{o} = (x, y, \phi)$ with x, y representing the world coordinates of the robot's position and ϕ denoting the heading of the robot. The restriction of the number of observation positions ensures that the variance of the place appearance is limited. Additionally, some kind of feature description model $\mathcal{M}$ of the place is added, where the nature of the description is variable. In this work, we use a contextual color histogram. Thus, the full description of a place is given by $P = (\mathbf{b}, \mathbf{O}, \mathcal{M})$.

B. *Color-based User Detection at Places*

The color-based feature model comprises the appearance of each place in multi-modal histograms. Each place is observed from different, but predefined view-points given different illumination conditions, e.g. ambient day-light and electric lighting in the evening. Therefore, the color model must be learned for each day-time and observation angle, independently. Instead of storing the histograms for each context (day-time, view-point) in a vector, a more efficient way is using a set containing only the observed appearances and the corresponding context. The size of this set can be limited by merging similar entries and keeping only distinctive ones. Therefore, we use a multi-modal color model augmented by a discrete context distribution capturing the circumstances of the histogram's acquisition.

1) Multi-modal Contextual Color Model: The model is defined by $\mathcal{M} = \{\kappa_1, \ldots, \kappa_n\}$, where $\kappa_i = (H_i, C_i)$ represents a component in the model with H_i denoting a color histogram and C_i being a multi-dimensional discrete context distribution. The histogram is 3 dimensional in RGB color space with 8 bins in each dimension (HSV and Lab color space showed no significant difference in performance). The context distribution captures arbitrary aspects of the origin of the histogram in separate dimensions. In this work we use view-point and day-time, which are represented by discretized Gaussians to account for slight variations in the aquisition. The mean of the view-angle Gaussian is given by the place's position $\mathbf{b}$ and the current observation position $\mathbf{o}_i \in \mathbf{O}$ (discretized into 9 bins) and sigma is set to 1.0. The mean of the day-time Gaussian is set to the current hour of the day with $\sigma = 1.5$. These values were determined empirically. Figure 3a displays the state of the two dimensional context distribution in two small lines above each color histogram. Red color indicates the probability of a state in the corresponding dimension. We set the maximum number n of components in $\mathcal{M}$ to 9. At the start of training, the model comprises zero components. At first a histogram is extracted from the box of the non-occupied place in the camera image and added as a new component to the model. Once the number of components exceeds n, the model must be pruned by merging similar components. This is done by first calculating the pairwise similarity s of all components:

$$s = \mathcal{BC}\left([H_i, C_i], [H_j, C_j]\right) \ , \tag{1}$$

where $[H, C]$ is the concatenation of the histogram distribution and the context distribution and $\mathcal{BC}(p, q)$ denotes the Bhattacharyya coefficient of two distributions:

$$\mathcal{BC}(p, q) = \sum_{x \in X} \sqrt{p(x) q(x)} \ . \tag{2}$$

The components with the highest similarity are merged by averaging the histograms and adding the context distributions. For example if two components have similar color histograms and are taken at similar time but from different view-points, the merged component represents the place for both view-points. The model $\mathcal{M}$ is learned for each place P in multiple teach runs including different day-times and illumination conditions. In the process of learning, the model maintains unique and distinctive representations of a place, but merges similar descriptions. Figure 3a shows an exemplary color histogram of a place on the couch. Each bin in the 3 dimensional histogram is plotted as a 2D area with its corresponding mean color with

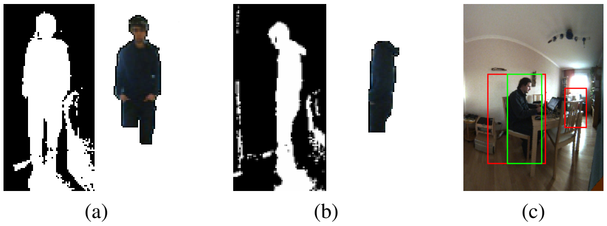


Fig. 4. User segmentation and sample detection. (a) Background subtraction output (binary image) and GrabCut refinement (color image) to learn a color model of the user. Shadow is removed very well and the segmentation is improved. (b) parts of the person are removed corrupting the segmentation. (c) Sample detection with place's bounding boxes (red), best fit correlation window (green). Smaller place to the right is not beeing checked because the robot is not on an observation position for that place.

the area size corresponding to the bin height. The histograms capture different lighting conditions (cf. Fig 3c), e.g. the couch normally appears in yellow-green (third column), bright given sunlight (second histogram in top row) or very dark at evening (histogram in the middle). Note that the model contains similar color histograms, but with different context distributions (first column).

2) Learning of the User Model: The color model of the user is similar to the aforementioned color model of places, but without the context distribution. Model learning is done by first creating a Gaussian Mixture background model [12], when the robot is standing still and no hypotheses are in front of the robot's camera (given by the tracker output). This background model is used for background subtraction once a hypothesis is visible in the image. To remove shadows and to refine the segmentation we apply the GrabCut algorithm [16]. The algorithm is automatically initialized with foreground pixels of the segmentation and background pixels in the bounding box of the person. A problem is the consistent segmentation of the user in the image. Although the GrabCut algorithm usually produces satisfying segmentation (Fig. 4a), from time to time background pixels are misclassified in the segmentation, or parts belonging to the person are left out (Fig. 4b). Therefore, at the moment and as a kind of interim solution, we trigger the learning of the user model once per day when the robot is standing in front of a white wall. The user is then asked to walk in front of the robot's camera. Figure 3b shows an exemplary learned color model of the user capturing mostly blue clothing which appear green under artificial light. With nine components, the user model can only represent a limited variety of different clothes. However, we observed that elderly people usually wear clothes in similar coloring. Yet, one drawback remains: the color of the user's clothes has to differ from the place's color, otherwise the method is likely to fail.

3) Recognition of the User: Once the place models and the user model have been trained, the system is able to detect the user in arbitrary poses at the learned places. For that purpose, the robot drives to the predefined observation positions and checks each place. By comparing the current appearance to the place and user model, the system decides if the place is occupied by the user. Therefore, the robot first extracts the current color histogram H_c from the place's box in the camera image. Furthermore, a context distribution C_c is created including current day-time and view-angle. The system now calculates the similarity of the current observation histogram H_c to the color histogram H_l of the place model using the Bhattacharyya coefficient:

$$s = \mathcal{BC}(H_c, H_l) \ , \tag{3}$$

where H_l is the histogram of the best matching component κ_l in the place model with l selected by:

$$l = \underset{i=1,\dots,n}{\arg\max} \left\{ \mathcal{BC}\left([H_c, C_c], [H_i, C_i]\right) \right\} \ . \tag{4}$$

Consequently, Eq. (3) is also used to calculate the similarity to the user model. Yet, a direct comparison of the complete histogram H_c to the user model's histograms would result in a very low match value, because the user usually only occupies a small region in the place's box and many background pixels are included in H_c. Therefore, the similarity to the user model is calculated by using a correlation window inside the place's bounding box and shifting it in a sliding window fashion to find the highest similarity. A possible way to determine the best size of the correlation window online would be to just try different sizes sequentially and use the one with the highest similarity. Yet, in this work, we used an empirically determined fixed size window with a width equal to 40% of the place's width while the height was kept (Fig. 4c). To select the best matching component κ_l from the user model, Eq. (4) is applied again, but in this case the context distribution is omitted and only the histograms are used.

If the user is present, this results in low similarity to the place model, because the appearance of the place is partially covered, and a high similarity to the user model, because the correlation window fits to the position of the user. If the user is not present, the results are vice versa. Proper decision criteria must be defined for both similarities to decide if a place is occupied. To this end, we trained a single linear Support Vector Machine (SVM) [17] on data of multiple labeled runs with empty and occupied places. The resulting SVM then decides for each place if the user is present given the similarities to both the place and user model. If the training data is diversified enough, the SVM is generally applicable to other scenarios with different place and user models without the need of retraining. We also tried to only use the similarity to the place model for decision making, which would obviate the need of a

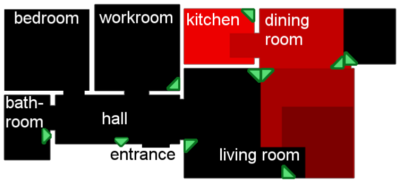
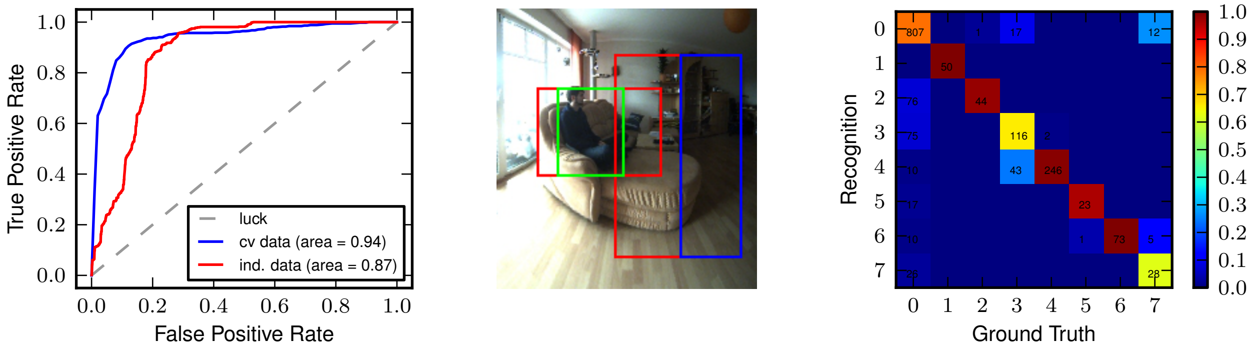


Fig. 5. Infrared sensor activation. Triangles indicate the positions of infrared motion sensors in the map of the apartment. Red color codes the time since the last movement sensor was activated by the user (brighter means more recent).



(a) ROC

(b) Overlapping places

(c) Confusion Matrix

Fig. 6. Evaluation of the place detection approach. (a) ROC curves for cross-validation and independent data. (b) Overlapping place's boxes (red) induce confusions. Correlation windows on person detection (green) and no detection (blue). (c) Confusion Matrix of multiple independent test runs. Classes $1-7$ represent different places, class 0 denotes user was not present.

user model. Unfortunately, the resulting performance was very poor compared to the results presented in Sec. VI. Another motivation of the two model approach is that it prevents false positive detections when objects, e.g. pillows or bags are left on the sofa, because they usually differ from the user model.

V. Integration of Infrared Motion Sensors

Every time the user needs to be sought in the apartment, because she or he got out of the range of the robot's sensors, a proper search strategy is needed. Therefore, the robot checks each of the aforementioned observation positions for the user's presence using the tracking framework including the place-detection module. The tracking system also detects a standing user while driving from one observation point to another. Generally, if no data is available from the motion sensors, the robot starts with the observation position closest to the last known position of the user.

By incorporating optional infrared motion sensors to the system, a more sophisticated search strategy can be applied. Each of the stationarily installed sensor fires in a 6-second-cycle when anybody is moving in a 90° area within a maximum range of about 4 meters in front of the sensor. The data is transmitted to the robot via Wi-Fi. By means of overlapping sensor areas, the spatial resolution can be slightly increased. Despite that, the achievable spatial accuracy of the sensors is not sufficient for fine user detection for interaction purposes, but more than enough to decide where to search first. The map shown in Fig. 5 is created by the robot using the most recent activation of each sensor. A history cue is built up for decision making. In the given example, a person has been sitting on the couch in the living room and moved to the kitchen where she or he is resting. On a given seek task, the robot starts with the place with the most recent activation (kitchen) and goes on to the ones with older activation (dining room, living room). If two observation positions yield the same time due to the large areas covered by the sensors, the closest one to the last known user position and the current position of the robot is used. When the robot arrives at a certain position, the place is marked as visited and it gets suppressed in the next selection cycle. By means of that selection algorithm, the complete apartment will be checked for the user. Generally, using the sensor information decreases seeking time enormously, because the robot usually drives directly to an observation position close to the user. Furthermore, the overall detection performance of the system is increased since the robot often instantly checks the right place occupied by the person lowering chance of false positives on empty places. This is experimentally examined in more detail in Sec. VI-B.

VI. Experiments

We separately evaluated the place detection module alone to detect lounging people (all other modules of Fig. 1 disabled) in Sec. VI-A and the complete tracking framework (all modules enabled) in Sec. VI-B.

A. Evaluation of User Detection at Places

We first learned the appearance of seven predefined (empty) places of the apartment (60 m^2, 3 rooms) in multiple training runs including three different lighting conditions – ambient day light, bright sunlight, and artificial light at evening. Furthermore, a multi-modal user model was trained with the user wearing two different clothes, which she or he also wore in the test runs. In the test scenario, the robot was placed on a fixed starting position and was searching for the user, who was either lounging at one of the places or not in the apartment. The robot checked each place for the user's presence and logged the similarities to the place and user models. The aforementioned linear SVM was trained and tested via 5-fold cross validation on the collected data of similarities of all places. The ground truth of the user's presence was labeled manually. For evaluation, we calculated the probability of the test examples belonging to the two classes of the SVM model (user present and not). By varying the probability threshold that is required to assign an example to one class, an ROC curve can be plotted (Fig. 6a). The blue curve shows the ROC of the cross-validated data used for training and validation. The red curve was generated on data from multiple independent test runs not seen by the system before. The high true positive rate and a low false positive rate of the red curve indicate that the system is actually able to robustly detect the user on the places. Furthermore, we evaluated the detection performance of the place detection system on the independent data sets used to generate the red ROC curve. The trained SVM is used to decide if the user is present or the place is empty. Each place is considered as one class and classification rates are calculated. Since the robot checks different places for the user's presence, it can wrongly detect the user at a place different from the ground truth. Additionally, sometimes more than one place is visible in the robot's camera image (Fig 6b). Hence the detection of places can be confused. Therefore, for evaluation a confusion matrix is chosen (Fig. 6c). Class 0 is used to denote that the user is not present (accumulated over all places). The classes $1-7$ correspond to places in the hall, at desk, 2 couch positions, 2 chair positions and an arm chair, respectively. Detection rates for each class are given in the main diagonal of the matrix. The average classification rate is above 85% with the biggest outliers in classes 3 (couch) and 7 (arm-chair). When the user is resting on the couch, he occasionally occupies two places (class 3 and 4) due to overlapping boxes (Fig. 6b). This results in high similarities to the user model in both couch places leading to the relative high confusion in class 3. In the case of class 7, direct sunlight caused the camera to overexpose and proper color extraction was hardly possible. This drawback, however, is inherent to all color-based approaches. Additionally, some false positive detections occurred (non-0-predictions of class 0).

B. Hide and Seek Scenario with Tracking Framework

We also tested the actual performance of the robot to localize the user in the apartment, once she or he got out of the robot's vicinity. Because the robot should deliver reminders and incoming video calls, it should robustly find the user as quickly as possible. For evaluation, we played several games of hide and seek a month after the aforementioned experiments without training new place models. We calculated the average search time and the success rate of over 100 different hide and seek games. In these games, the tracking system of the robot applied all detection modules shown in Fig. 1. Unless stated otherwise, the infrared sensors were used to sequence the observation positions as described in Sec. V.

TABLE I
RESULTS OF DIFFERENT HIDE AND SEEK SCENARIOS.

	games	suc. games	suc. rate	avg. time
(a) lounging/standing	73	54	0.74	27.8 s
(b) standing	15	13	0.87	32.2 s
(c) w/o motion sensors	19	8	0.44	37.2 s
(d) user not in apart.	18	3	0.15	25.4 s

Each game (or test run) started with the robot situated on a fixed starting location. The user then "hid" somewhere in the apartment by resting on one of the learned places. Occasionally, the user did not occupy any of these places, but stood somewhere in the apartment. Furthermore, in a few games the user was not present at all. The ground truth of the user's position was labeled manually. The robot then started searching for the user by driving to each observation position and checking for the user's presence. The initial observation position was selected by using the output of the infrared sensors. If the user was not found on a specific place, the robot went on checking the other places. If the robot found the user lounging at a resting place or standing somewhere in the apartment, it logged the detection position and the time at which the detection was made and returned to the starting location to end the game. Once the user moved to another location, a new game was started. If the user was not present in the apartment or the robot failed to detect her or him on the specific place, the robot returned to the starting location after checking all places, logging the moment of arrival. Table I shows the results of these experiments. We regarded a game as successful if the Euclidean distance of robot's detection to the ground truth was below 1 m or if the robot returned to the starting location after checking all places, if the user was not present. We calculated a success rate by dividing the number of successful games by the total number of games. Table I(a)-(c) summarizes independent games, where the user was always present somewhere in the apartment. Tab. I(a) depicts games in which the user was mostly lounging at different places and occasionally just standing. The success rate was rather high with over 0.70. Errors mostly occurred when the user was in an unfavorable position (cf. Sec. VI-A and Fig. 7) and proper histogram extraction was impossible. Furthermore, if the latest active smart home sensor covered multiple observation positions, the robot might check some places before the right place increasing the chance of falsely detecting the user on the wrong place. The average time over all successful and unsuccessful games was 27.8 seconds with a minimum of 5 and a maximum of 122 seconds (only 2 search runs took longer than 60 s). We also tested the performance of the system in addtional games, in which the user was always standing somewhere in the apartment (Tab. I(b)). Compared to the aforementioned games, where the user usually lounged at a place, the success rate increased to 0.87. This is because the range-based and HOG-based detection modules are particularly dedicated detecting standing people. The average time to find the user increased a little, because the user was not on one of the predefined places but had to be found by the robot when driving from one observation point to another in the perception area of the IR-sensor activated as the last. When disabling the infrared motion sensors (Tab. I(c)), the success rate dropped down to 0.44 and the average time to finish a game increased from 32 to 37 seconds. Without the initial hint of the IR-sensors, the robot had to check each place, increasing the average search time and the chance of false positives. From the findings of Sec. VI-A, we assume the probability of a correct classification for a single place to be $p(c) = 0.80$. Since, the robot checks multiple places, the probability of correct classification is given by: $p(x) = p(c)^n$, where n is

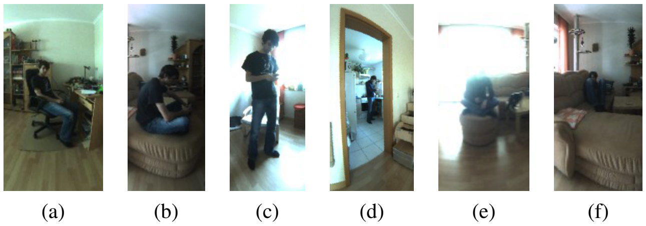


Fig. 7. Hideout examples. Easy hiding places: (a), (b), and (c). Difficult hiding places: (d) great distance, (e) overexposure, and (f) small area in place.

the number of places visited and $p(c)$ is the mean classification rate for one place. Hence, the more places the robots needs to check before reaching the place occupied by the user, the lower the probability of a successful game. This becomes extreme, if the user is not at home, in which case the robot should check all places and return to the starting location. Given the seven places in our apartment and a mean classification rate of 0.80 for each place, the expected probability of a successful run is only 0.21. This explains the low success rate when no person is present in the apartment (Tab. I(d)). Ideally, the average time of these games should be the duration it takes the robot to check all places in the apartment and return to the starting location. In our experiments the average time over all games was only 25 seconds. The reason for that is that the aforementioned false detections mostly occurred on places checked early by the robot. To raise the success rate, one could increase the influence of the IR-sensors. If one assumes that the sensor hint is very certain, the robot could only check the area of the latest sensor activation for the user's presence. Hence, only a small number of places would have to be checked lowering the chance of false detections. Furthermore, if the motion sensors were assumed to be very reliable, the robot would not need to check for the user at all if no sensor is active. Integrating user feedback could further diminish the problem of false positives. When detecting a user at a place, the robot could verify its estimation by prompting the user for a feedback. This may lead to some scenarios where the robot talks to empty chairs, but the overall robustness would be highly increased. Yet, the problem remains if the user should be observed passively and silently.

VII. Conclusion and Ongoing Work

We presented a tracking framework for mobile robots to detect people in home environments. Besides the integration of state-of-the art detection modules in a real-time capable framework, a novel method for detecting lounging people independent of their resting pose at predefined places was presented. The idea of the method is to learn a color-based appearance model of the user and predefined places in the apartment, beforehand. Then SVMs are trained to decide if a place is occupied by the user. Afterwards, the system is able to autonomously detect the user in the given environment. Like most color-based approaches the method assumes that the color of the user's clothes differs from the color model of the places. Experiments on multiple independent test runs substantiate that the approach actually improves person detection performance in living environments by detecting the user also in situations not captured by state-of-the-art detection and tracking systems on mobile robots. The performance of the tracking framework to continuously observe and find the user in the apartment was tested in over 100 hide and seek games. Assisted by infrared motion sensors, the robot was able to correctly find the hidden user in more than 70% of all games. In future work, we want to replace the manual definition of places by an interactive training guided by the user. Furthmore, the robot could learn which observation position is most suitable to robustly detect the user on a specific place. Additionally, we want to increase the success rate to find the user in the apartment by limiting the search to those areas given by the infrared motion sensors or incorporating user feedback to verify the detection. Last but not least, we are working on a HOG-based [3] representation of each place to be unsusceptible against illumination and color changes.

References

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

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

[3] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Conference on CVPR*, 2005, pp. 886–893.

[4] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models." *IEEE TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[6] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3D human pose annotations," in *IEEE 12th ICCV*, 2009, pp. 1365–1372.

[7] B. Leibe, A. Leonardis, and B. Schiele, "Robust Object Detection with Interleaved Categorization and Segmentation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 259–289, Nov. 2007.

[8] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis, "Human detection using partial least squares analysis," in *IEEE 12th International Conference on Computer Vision*, 2009, pp. 24–31.

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

[10] H.-M. Martin, Chr., Schaffernicht, E., Scheidig, A., Gross, "Sensor Fusion using a Probabilistic Aggregation Scheme for People Detection and People Tracking," *RAS*, vol. 54, no. 9, pp. 721–728, 2006.

[11] S. Müller, E. Schaffernicht, A. Scheidig, B. Hans-Joachim, and M. Gross-Horst, "Are you still following me," in *Proccedings of the 3rd European Conference on Mobile Robots*, 2007, pp. 211–216.

[12] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999, pp. 246–252.

[13] T. Han and J. Keller, "Activity Analysis, Summarization, and Visualization for Indoor Human Activity Monitoring," *Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1489–1498, 2008.

[14] R. B. Rusu, J. Bandouch, F. Meier, I. Essa, and M. Beetz, "Human Action Recognition Using Global Point Feature Histograms and Action Shapes," *Advanced Robotics*, vol. 23, no. 14, pp. 1873–1908, 2009.

[15] C. Schroeter, A. Koenig, H.-J. Boehme, and H.-M. Gross, "Multi-Sensor Monte-Carlo-Localization Combining Omnivision and Sonar Range Sensors," in *Proc. of the 2nd ECMR*, 2005, pp. 164–169.

[16] A. Rother, Carsten and Kolmogorov, Vladimir and Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM SIGGRAPH 2004 Papers*. ACM, 2004, pp. 309–314.

[17] C.-C. C. Lin and Chih-Jen, "LIBSVM: a library for support vector machines." 2001. [Online]. Available: http://www.csie.ntu.edu.tw/˜cjlin/libsvm