Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Finding misplaced items using a mobile robot in a smart home environment*

Qi WANG[1], Zhen FAN[1], Wei-hua SHENG[2], Sen-lin ZHANG[1], Mei-qin LIU†‡[1]

[1]*College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*

[2]*School of Electrical and Computer Engineering, Oklahoma State University, Stillwater OK 74078, USA*

†E-mail: liumeiqin@zju.edu.cn

Received May 3, 2018; Revision accepted July 30, 2018; Crosschecked Aug. 9, 2019

Abstract: Smart homes can provide complementary information to assist home service robots. We present a robotic misplaced item finding (MIF) system, which uses human historical trajectory data obtained in a smart home environment. First, a multi-sensor fusion method is developed to localize and track a resident. Second, a path-planning method is developed to generate the robot movement plan, which considers the knowledge of the human historical trajectory. Third, a real-time object detector based on a convolutional neural network is applied to detect the misplaced item. We present MIF experiments in a smart home testbed and the experimental results verify the accuracy and efficiency of our solution.

Key words: Home service robot; Smart home; Heterogeneous sensors; Autonomous robot retrieval
https://doi.org/10.1631/FITEE.1800275 **CLC number:** TP242.6

1 Introduction

1.1 Motivation

These days, smart homes have greatly improved the quality of our life. Intelligent service robots, for example, have significantly reduced our workload by undertaking some tasks such as vacuum cleaning. In recent years, the growth of the elderly population has called for more service robots for elder care. Older adults may forget things they have done due to their declining memory (Choi et al., 2009). Each time they want to locate a specific item, it might be a problem for them to recall where they put it after its latest use (Meng and Lee, 2006).

Hence, an autonomous robot retrieval system (ARRS) (Ahern et al., 2015) is highly desirable to search for misplaced items in home environments. Traditionally, approaches for locating misplaced items use wireless tags to track the position of the items (Tesoriero et al., 2009; Huynh et al., 2014; Schwarz et al., 2014; Ravi et al., 2016). However, the lifetime of the tags is limited due to the use of batteries. An alternative approach for misplaced item finding (MIF) is visual search (Sadhu et al., 2013; Xu et al., 2016). Robots move around the environment to search for the misplaced item while cameras mounted on the robots acquire scene images during their movement. An object detector can be developed on the robot that is capable of locating the misplaced item in a scene image. To achieve this, it is a crucial task to generate an appropriate running path. It is better if the robot knows where to search for the misplaced item and how to go to that area, which can be regarded as a problem of robot path planning. In our previous work

‡ Corresponding author

* Project supported by the Basic Public Research Program of Zhejiang Province, China (No. LGF18F030001) and the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT1800414)

ORCID: Mei-qin LIU, http://orcid.org/0000-0003-0693-6574

(Wang et al., 2017), a back and forth motion pattern was implemented for the robot to search for the misplaced item. However, a simple back and forth motion is neither accurate nor efficient. A complete and exhaustive search of every room is very time-consuming for the robot. Therefore, we need to introduce some complementary information to guide the robot's search movement, which can be provided by the smart home environment.

From our daily experiences, we know that the location of a common item and human movement are highly correlated, because the location of an item is usually changed by human use. Imagine a scenario in which an older adult has misplaced his/her key after entering a room. Under the premise that the person had the key when he/she opened the door, it should be certain that the key is most likely placed at a spot close to the location where the user has visited recently. Thus, we can assume that the user drops things with a uniform probability while he/she is walking around.

In this study, we propose a new MIF solution for a home robot that relies on the historical information of human movement provided by the smart home environment. This study has three contributions. The first contribution is that we apply a data fusion method between a wearable sensor and multiple distributed movement detection sensors to estimate the historical trajectory of a resident. The human trajectory can be calculated by the smart home environment, instead of the robot. The second contribution is that we develop a robot path-planning method in which a preferable robot search path can be generated using knowledge of the human historical trajectory data. The third contribution is that our work demonstrates that smart homes can provide the much-needed context information for home service robots to better serve people.

1.2 Related work

In recent years, researchers have developed several solutions that use environmental information to search for misplaced items. Fujii et al. (2007) analyzed human behaviors in terms of misplaced items inside the home and introduced an item search service to assist in finding misplaced items. In their work, the item was recorded as a potentially misplaced item when the human-item relationship was changed by an interruption event. Active radio frequency identification (RFID) tags were attached to both the user and the items, and the positions of misplaced items could be obtained by the relative distances between them. Komatsuzaki et al. (2011) proposed an item search system called IteMinder using passive RFID tags and an autonomous robot. Known locations equipped with passive RFID tags served as reference points and provided a map-based search cue for the robot. The drawbacks are the long required time for full-environment scans and the limited lifetime of the tags as we mentioned before. Ekvall et al. (2007) proposed a solution in which simultaneous localization and mapping (SLAM) works in collaboration with object detection to generate an augmented metric map, and the robot detects items in the environment and places annotations in the map. In this way, robots can use the annotation information to reason about the relationships between items and places while retrieving a certain item from a particular room. However, a recently misplaced item may not be reflected in the annotated map, and therefore the robot may not be able to find the item according to the map. Rogers and Christensen (2013) presented a common sense-based path planner for item search tasks, using room connectivity and object information as the context. In their method, an object-in-room affinity model was developed by analyzing sentences in the common sense database, while contextual cues were learned from the training data and applied to guide robot movement in an item search task. Kollar and Roy (2009) used object-object and object-scene relationships inherent in the environment to develop a probabilistic model over possible object locations, which indicates that some items tend to co-exist in certain kinds of places. Using the items and scenes that the robot already knew, their model was capable of predicting the location of misplaced items. However, all the known items must be manually labeled before developing the model, which is a heavy burden on the user.

In reviewing previous research, we noticed that most researchers developed a probabilistic model that considers known object positions or common sense as the knowledge when planning the robot search movement. Therefore, the search path for a specific object might be fixed. Moreover, totally relying on the sensors on the robot to conduct the search consumes a significant amount of the robot's

resource. Therefore, it is desirable to develop a new MIF approach using the contextual information provided by the smart home environment.

2 Overall concept

A robotic MIF system is expected to take advantage of the services provided by the smart home environment. Human location tracking is such a service. In this section, we propose the overall concept of the robotic MIF system.

The proposed approach is to use human historical trajectory data from the smart home environment to support MIF. We present the conceptual design of the robotic MIF system (Fig. 1), which consists of mainly four parts: (1) a mobile service robot which is equipped with various sensors for visual search purposes; (2) a human localization sensor network which localizes the human in an indoor environment to provide the human historical trajectory data; (3) a user interface which allows the user to easily interact with the robot; (4) a robot path planner which uses human historical trajectory data to generate a preferable search path that minimizes search time. These four parts are connected to the local area network to offer fast communication among them.

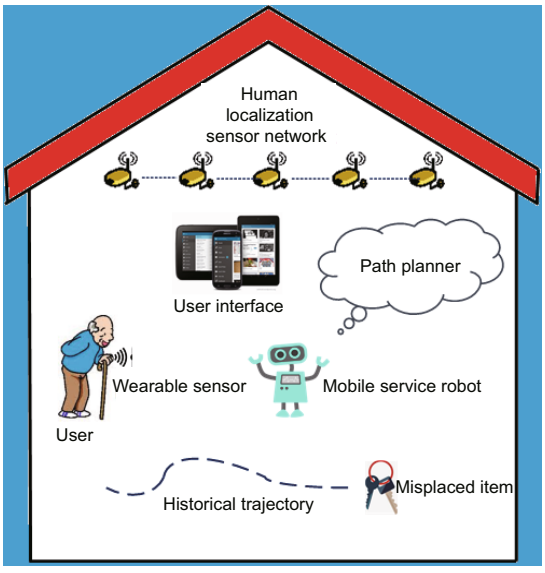

Fig. 1 Concept of the robotic misplaced item finding (MIF) system

The flowchart of our MIF procedure is shown in Fig. 2. After the user enters the room, the human localization network and the wearable sensor work together to estimate the user's trajectory. Once the user asks for a certain misplaced item, the user interface converts the user's voice command into text, which is recognized as a start-searching command to initiate the MIF procedure. The item-related trajectory data are chosen from the recorded historical trajectories and then streamed to the robot, where these data are used to generate a search path by the path planner. Real-time scene images collected by the robot are passed to an object detector to determine if there is a misplaced item in any of the images. The location of the detected item will be determined and a voice notification will be generated through the user interface once the misplaced item is found.

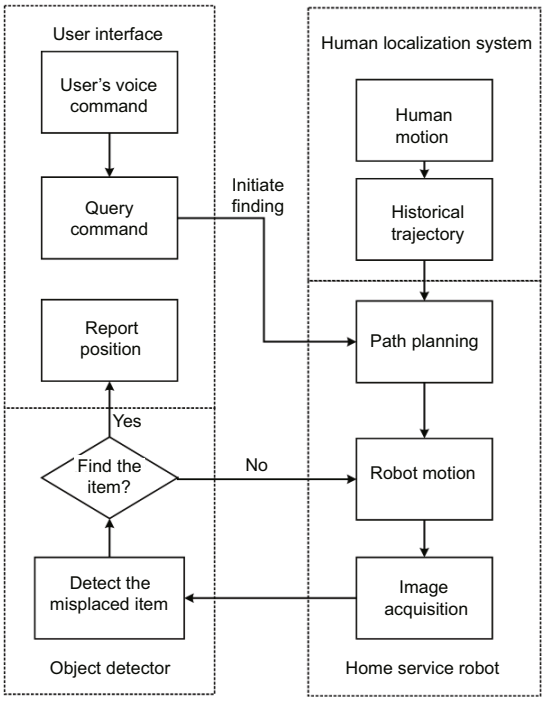

Fig. 2 Misplaced item finding (MIF) procedure

3 Hardware platform

In this section, we describe the main components of the robotic MIF system in detail, including the service robot, the smart home, and the user interface.

3.1 Hardware setup for the service robot

The mobile service robot we have designed for the MIF task in a home environment is shown in Fig. 3. The robot consists of a mobile robot base,

a laptop, and several peripheral sensors. The robot base is the TurtleBot platform, which is a safe, reliable, robust, and feature-rich mobile chassis. The computation engine of the robot is a laptop equipped with a dual-core processor, which runs the Robot Operating System (ROS, 2018). For perceiving objects in the environment, we equipped the robot with various sensors. A fixed-focus RGB-D camera captures the scene images necessary for object detection. A laser range finder (LRF) mounted on the top of the mobile base measures the obstacles for mapping and localization purposes.

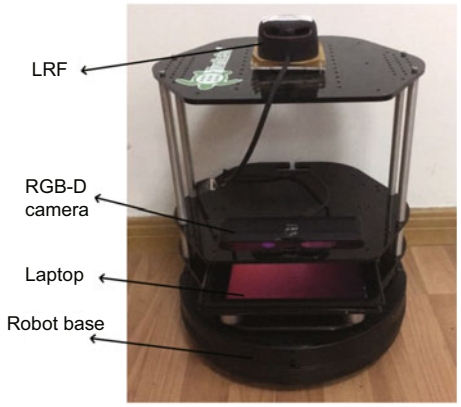

Fig. 3 Mobile service robot

3.2 Hardware setup for a smart home environment

3.2.1 Smart home testbed

Our smart home testbed was established in a 5.72 m × 6.85 m (19 ft × 22 ft) room. It simulates a home environment that consists of a living room, a bedroom, a kitchen, and a bathroom. Fig. 4 shows an example of the smart home testbed.

3.2.2 Passive infrared sensor network

A passive infrared (PIR) sensor is an electronic sensor that detects infrared light radiated from moving objects (PIR, 2018). The PIR sensor used in this system is the Panasonic EKMC1601111, which is capable of detecting human movement when the resident is in its field of view (FoV). Fig. 5 shows one PIR sensor node. The PIR sensor is connected to an Arduino microcontroller board. An XBee shield, along with the XBee module, is mounted on the Arduino board, which is used to transmit the sensor

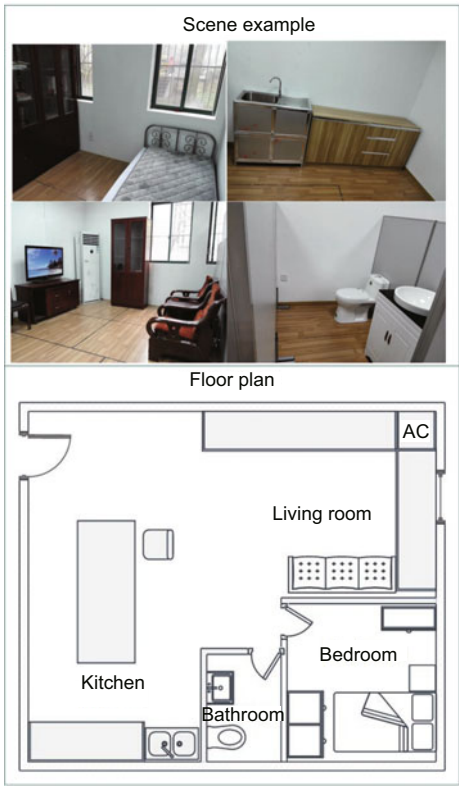

Fig. 4 An example scene of the smart home (top) and the floor plan (bottom)

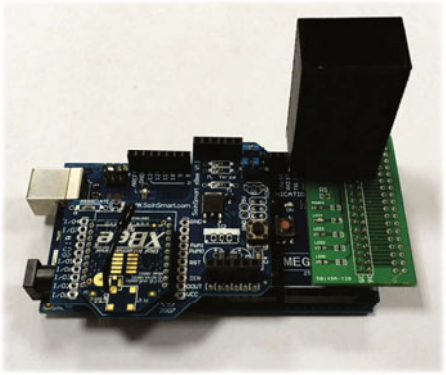

Fig. 5 A passive infrared (PIR) sensor node

data over the ZigBee protocol to the gateway node. A black cover is attached around the PIR sensor to limit the sensor's FoV. The PIR sensor network consists of seven PIR sensors in total.

3.2.3 Wearable sensor node

We used the wearable sensor node SensorTag v2 from TI (http://www.ti.com/lit/ug/tidu862/tidu862.pdf). The SensorTag operates as a bluetooth low-energy (BLE) peripheral slave device based on

the low-power and high-performance CC2650 multi-standard wireless microcontroller unit (MCU) platform. Inside the MCU platform, there is a nine-axis motion MPU-9250 sensor (inertial measurement unit, IMU) that provides the motion data including three-dimensional (3D) accelerations, 3D angular velocities, and 3D magnetic field data. The CC2540 universal serial bus (USB) dongle acts as a central device (BLE master). It connects to a Windows PC USB port and is preloaded with necessary software to collect the sensor data. This sensor node is attached to the left side of the user's waist (Fig. 6).

3.3 User interface design

To provide a friendly interface for the user, we integrated an Android application (Fig. 7) as a remote control interface. It can display the real-time positions of the robot and the resident on the home floor plan. The speech-to-text service offered by the Xunfei online voice recognition platform is also implemented in this application (Xunfei, 2018). It enables the robot to understand the user's voice command when querying a certain item and reports the search results.

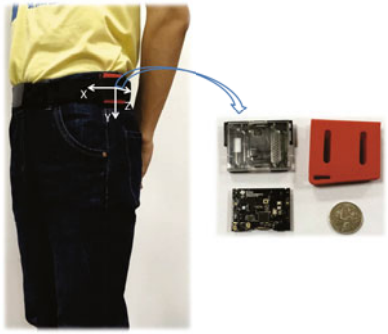

Fig. 6 Wearing position (left) and wearable sensor node (right)

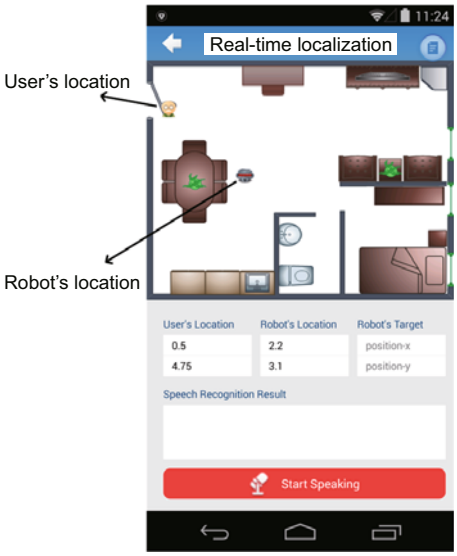

Fig. 7 Remote user interface

4 Methodology

In this section, we discuss the detailed methodology for the proposed robotic MIF system, including human location estimation, search path planning, and vision-based object recognition.

4.1 Human location estimation

One feature of our system is the ability to obtain human trajectory information from the sensor data. As illustrated in the hardware setup for the smart home environment in Section 3, the motion data from a wearable sensor are used to develop the user's motion model. Meanwhile, the PIR sensor network collects human movement data as external observations. We developed a particle filter based framework to fuse the information from these two channels to obtain an accurate human location estimate. The estimation process is depicted in Fig. 8.

4.1.1 Motion model

A wearable sensor can be used to develop the motion model of the user. By implementing the pedestrian dead reckoning (PDR) algorithm (Susi et al., 2013) in our system, we can estimate a coarse human location through the measurement of movement distance and heading. Generally, the PDR

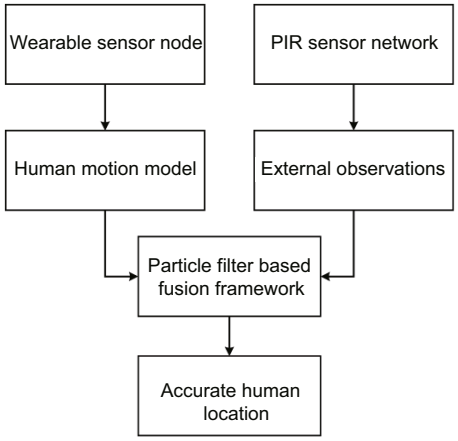

Fig. 8 Location estimation process

algorithm computes the movement distance by counting the user's walking steps, while the walking step detection can be considered an acceleration detection problem. Here we use X-axis acceleration (a_x), Y-axis acceleration (a_y), and the acceleration magnitude

$$a_v = \sqrt{(a_x)^2 + (a_y)^2 + (a_z)^2} \tag{1}$$

from the wearable sensor as raw signals. These raw signals are first filtered by a low-pass filter to remove noise. Then features are extracted using a sliding window size of 32 with an overlapping size of 16 between two consecutive windows. At a sampling rate of 10 Hz, each sliding window represents the data of 3.2 s, which is sufficient to capture the cycles in activities like walking. An adaptive-threshold-based algorithm is performed in acceleration detection of each channel to adapt to different users and different walking modes, in which the signal valley threshold (T_v) and the signal peak threshold (T_p) can be calculated as follows:

$$T_\mathrm{v} = \frac{V_k + V_{k-1} - 1}{2} + \left(K - \frac{V_k + V_{k-1} - 1}{2}\right) C_1, \tag{2}$$

$$T_\mathrm{p} = T_\mathrm{v} + \sqrt{K - T_\mathrm{v}} C_2, \tag{3}$$

where K is the limiting condition to renew the threshold, satisfying

$$K = \begin{cases} P_k, & P_k < P_{k-1}, \\ P_{k-1}, & P_k \geq P_{k-1}. \end{cases} \tag{4}$$

Here, P_k and V_k are the maximum and minimum acceleration values of walking step k respectively, P_{k-1} and V_{k-1} are of walking step $k-1$, and C_1, C_2 are the parameters determined experimentally. We consider it a walking step if both the peak and valley are detected within a sliding window.

Meanwhile, we use the angular speed to estimate the heading angle change when the user is walking by integrating angular speed over time. The heading angle change can be obtained as

$$\theta_k = \theta_{k-1} + \Delta\theta, \tag{5}$$

$$\Delta\theta = \int_{t-1}^{t} \omega_t \mathrm{d}t, \tag{6}$$

where $\Delta\theta$ is the accumulated heading angle change between walking steps $k-1$ and k. Note that $t-1$ and t are the starting points of walking steps $k-1$ and k, respectively.

Combining the walking steps with the heading angle, we derived the human motion model as follows:

$$\boldsymbol{L}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + d_k \cos\theta_k \\ y_{k-1} + d_k \sin\theta_k \end{bmatrix} + \boldsymbol{n}_k, \tag{7}$$

where $\boldsymbol{L}_k$ are the coarse location coordinates at walking step k, d_k is the walking step length which remains constant for a certain user, and $\boldsymbol{n}_k$ is the process noise.

4.1.2 Human location based on PIR observations

In our smart home testbed, we distributed seven PIR sensors on the ceiling without overlapping coverage, ensuring that the majority of the room can be covered by the sensors. The installation of the PIR sensors and their corresponding covering areas are shown in Fig. 9. Each PIR sensor can detect human movement when the user is entering or leaving its FoV. Based on these characteristics, we established a queue of triggered PIR identifiers. Once a new

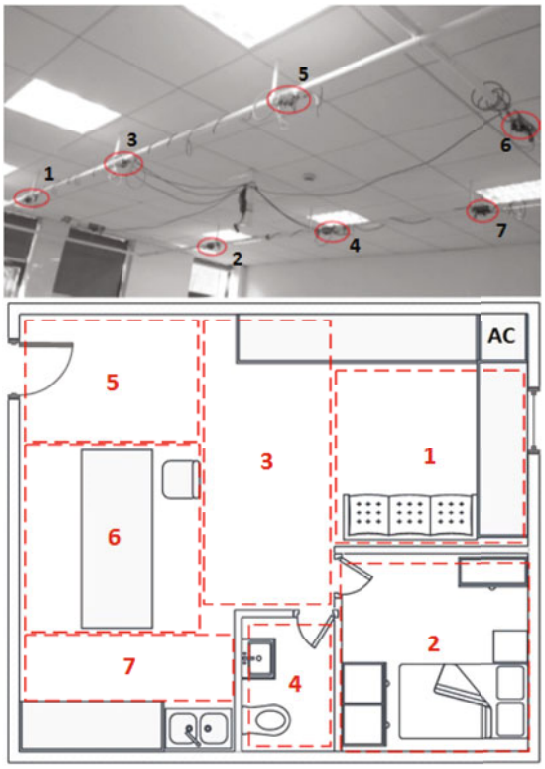

Fig. 9 Passive infrared (PIR) sensor network installation (top) and sensors' corresponding covering areas (bottom)

PIR sensor detects the user's movement, its identifier is put into the queue and the user's movement information is used to calculate the user's location. The foremost identifier is moved out of the queue a certain time after the last trigger event, so the sensor can be retriggered. The trigger information of an identifier in the queue is not used again until the previous one is moved out. Thus, the measurement model of the PIR sensor can be defined as follows:

$$P(z_k^{\mathrm{PIR}}=1|\boldsymbol{L}_k^i)=\begin{cases}1, & \boldsymbol{L}_k^i\in \text{ PIR sensor's FoV},\\ 0, & \boldsymbol{L}_k^i\notin \text{ PIR sensor's FoV},\end{cases} \tag{8}$$

where $P(z_k^{\mathrm{PIR}}=1|\boldsymbol{L}_k^i)$ is the likelihood function of location $\boldsymbol{L}_k^i$, and $z_k^{\mathrm{PIR}}=1$ represents the PIR sensor being triggered. $\boldsymbol{L}_k^i$ may be one of two conditions: the user is either in the same room or in a different room that the particle represents.

4.1.3 Human location based on activity statuses

The indoor activity area of a user is limited by the layout of the room. Generally, human daily activities take place at specific locations, such as chairs for sitting, living room for walking, and bed for lying. Fig. 10 shows the layout of the room, in which the white, blue, red, and green areas represent the free space, unreachable obstacles, chairs, and the bed, respectively. Here, we use a_k to represent several different kinds of human daily activities such as walking, standing, sitting, and lying. Thus, the measurement model of human activity recognition can be defined as follows:

$$P(a_k|\boldsymbol{L}_k^i), \tag{9}$$

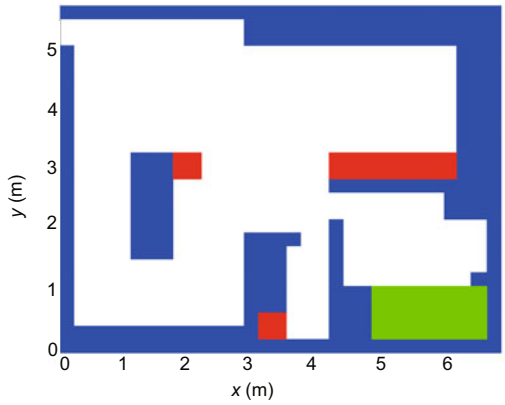

Fig. 10 Room layout (References to color refer to the online version of this figure)

where $P(a_k|\boldsymbol{L}_k^i)$ is the likelihood function of location $\boldsymbol{L}_k^i$. The distribution of $P(a_k|\boldsymbol{L}_k^i)$ is shown in Table 1.

Moreover, different activities show different patterns in their corresponding acceleration information (Fig. 11). Thus, the user's activity type can be recognized based on the analysis of the acceleration observations (Fig. 12).

Table 1 Distribution of $P(a_k|L_k^i)$

Area	$P(a_k\|\boldsymbol{L}_k^i)$			
	Standing & walking	Lying	Sitting	NaA
Free space	0.97	0.01	0.01	0.01
Obstacles	0.01	0.01	0.01	0.97
Chairs	0.01	0.01	0.97	0.01
Bed	0.01	0.64	0.34	0.01

NaA: not an activity

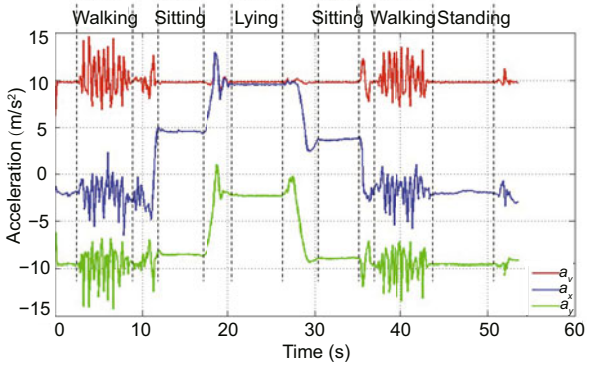

Fig. 11 Acceleration information of different activities

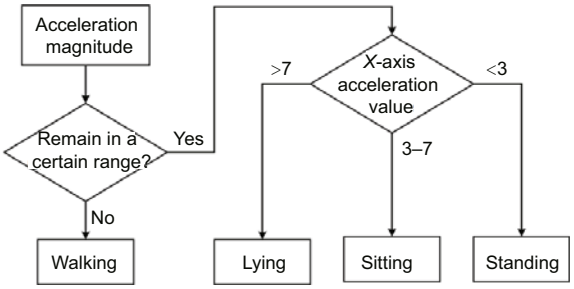

Fig. 12 Activity recognition procedure

4.1.4 Sensor fusion for localization

Based on the above analysis, we can accomplish accurate location estimation by fusing these data using a particle filter based fusion framework. In this framework, each particle represents a possible human

location associated with a particle weight, which indicates its probability of being the real human location. The propagation of particles is based on the human motion model as expressed by Eq. (7), while the particle weight ω_k^i is updated by the related likelihood as follows:

$$w_k^i = w_{k-1}^i p(z_k^{\mathrm{PIR}}|\boldsymbol{L}_k^i)p(a_k|\boldsymbol{L}_k^i). \tag{10}$$

The particle weight is normalized by the following equation after every update:

$$\omega_k^i = \omega_k^i \Big/ \sum_{i=1}^{P} \omega_k^i. \tag{11}$$

To avoid weight collapse in the filtering algorithm, a resampling procedure is implemented if all particles degenerate to a relatively ineffective level. We use the number of effective particles, N_{eff}, as the trigger for resampling, which can be calculated as

$$N_{\mathrm{eff}} = 1 \Big/ \sum_{i=1}^{P} (\omega_k^i)^2. \tag{12}$$

If N_{eff} is smaller than a threshold, the resampling procedure will be executed to concentrate particles in a higher-probability area.

Finally, the accurate human location can be estimated based on the mean of the particles' locations as follows:

$$\tilde{\boldsymbol{L}}_k = E(\boldsymbol{L}_k) = \frac{\sum_{i=1}^{P} \boldsymbol{L}_k^i \omega_k^i}{\sum_{i=1}^{P} \omega_k^i}. \tag{13}$$

The particle filtering algorithm for human localization is described in Algorithm 1.

Algorithm 1 Fusion algorithm for localization and tracking

1: Initialize P particles with location vector $\boldsymbol{L}_k^i$, heading θ_k^i, and weight $\omega_k^i = 1/P$ $(i = 1, 2, \ldots, P)$
2: Recognize human activity a_k and read z_k^{PIR} from the PIR sensor network
3: **if** A walking step k is detected **then**
4: Estimate walking step length d_k and heading angle θ_k. Propagate the particles according to Eq. (7) // prediction step
5: Update the weights of the particles according to Eq. (10) // update step
6: **if** $N_{\mathrm{eff}} < N_{\mathrm{t}}$ (N_{eff} is calculated by Eq. (12) and N_{t} is the judgment threshold) **then**
7: Implement the resampling procedure
8: **end if**
9: Estimate human location according to Eq. (13)
10: **end if**
11: Go to step 2

4.2 Search path planning

The next problem for our robotic MIF system is how to use the human historical trajectory information to compute a preferable search path that minimizes the expected search time of the misplaced item. After the environmental global map was created by the SLAM algorithm (Castellanos et al., 2001), we converted the global map into a grid map, which could be divided into several regions later. To generate a preferable search path, we used a modified genetic algorithm that considers the prior human trajectory information given in the previous subsection.

4.2.1 Map quantization and region partitioning

The global map created by the SLAM algorithm contains the environmental information for robot navigation. To make path planning solvable, the global map is converted into a grid map. Our smart home testbed was modeled as a 26×31 grid map with 22 cm×22 cm grids (Fig. 13a), in which black grids represent the obstacle area and white grids show the free space where the robot can navigate.

To reduce the time and energy consumption of the MIF procedure, the grid map is divided into regions and the robot should find the best sequence to visit the different regions. In each region, the robot rotates itself and conducts a visual search for the misplaced item. Thus, the visual search range of the robot must be considered when dividing the grid map, because the item may be out of the camera's maximum detection range or be occluded by the surrounding obstacles (Lin et al., 2015). Taking these factors into account, we segmented the grid map into 10 regions, A–J (Fig. 13b). In each region, the cell marked with the letter indicates the location where the robot rotates itself and conducts the visual search.

4.2.2 Integrating knowledge and path planning

Given the knowledge of the human historical trajectory and region partition, the crucial task for search path planning is to determine the probability

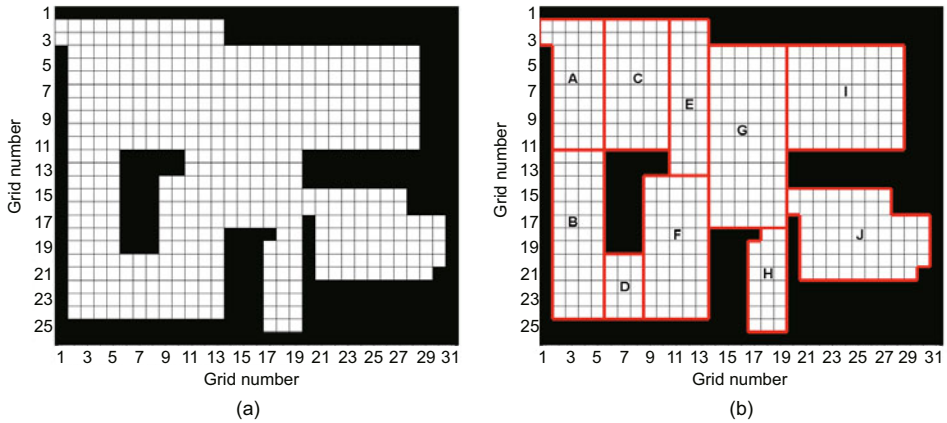

Fig. 13 Grid map (a) and region partition result (b)

of the misplaced item being located in a certain region, in other words, to generate a preferable region transition sequence in which a region with a high probability may have a better chance of being the first search area. A direct approach for the sequence planning problem is the direct search method. The robot checks all the possible region transition sequences and finds the best sequence directly. However, the number of possible region transition sequences grows by a factorial with the number of partitioned regions, which may cause a heavy computational burden on the robot. Here we adopt the genetic algorithm (GA) to solve this sequence planning problem. GAs are commonly used to generate high-quality solutions to optimization problems. The main idea of GA is to produce new solutions or visiting sequences by combining different candidate solutions, which should perform better than the previous generation.

In our current method, the conventional GA was modified by developing a new fitness function to solve the planning problem. After numbering regions A–J in the grid map to genes 0–9, each region transition sequence is represented in the form of a chromosome. For example, the region transition sequence C → A → J → E → H → G → I → F → B → D can be represented by a chromosome [2, 0, 9, 4, 7, 6, 8, 5, 1, 3]. To ensure the accuracy and efficiency of the MIF procedure, we constructed a comprehensive fitness function which considers the traveling time, the rotation time, and the probability of the region:

$$f=\sum_{n=1}^{N-1}\frac{D(s_{n-1},s_n)/v+T(s_{n-1},s_n)/\omega}{\rho_{s_n}}, \quad (14)$$

where N is the total number of regions in a given region transition sequence, $D(s_{n-1},s_n)$ and $T(s_{n-1},s_n)$ denote the moving distance and the rotation angle between two consecutive regions respectively, and v and ω represent the robot's linear and angular speed respectively.

Considering our assumption that the user drops things with the uniform probability while he/she is walking around, the probability ρ_{s_n} is dependent on the human historical trajectory knowledge derived in the previous subsection, which can be calculated as follows:

$$\rho_{s_n}=\frac{L_{s_n}}{\sum_{n=0}^{N-1}L_{s_n}}, \quad (15)$$

where L_{s_n} is the number of human trajectory points contained in region s_n. Note that if there is no trajectory point in this region, the corresponding probability ρ_{s_n} will be set to a small value ϵ.

The fitness function of our modified GA was optimized to generate a preferable region transition sequence with respect to the assumption that the user drops things with the uniform probability while he/she is walking around.

4.3 Vision-based object recognition

The proposed vision-based object recognition architecture for MIF is illustrated in Fig. 14, which consists of mainly two modules: one is for image acquisition and runs on the robot, and the other is for image processing and runs on a workstation. The workstation is equipped with an Intel Xeon E5-2603 quad-core processor at 1.8 GHz and an

NVIDIA Tesla K40 GPU (graphics processing unit). To identify the misplaced item within the scene images, our workstation includes an integrated fast object detector, YOLOv1 (Redmon et al., 2016), which is based on a pre-designed convolutional neural network (CNN). The robot collects the RGB scene images from its camera and sends them to the workstation for further processing; the workstation is responsible for item detection and sends recognition results to the robot.

The CNN that we used is illustrated in Fig. 15; it generates three types of outputs from the scene images: the class probability, the bounding box coordinates, and a confidence value. Based on these outputs, we designed an evaluation flowchart to identify if there is a misplaced item within the scene images (Fig. 16), which generates two classes: "item" and "no item." The class probability indicates if there is any pre-trained item in the scene images. If a specific item is detected, its corresponding confidence and bounding box coordinates are checked for further confirmation. The misplaced item is considered detected if and only if all the object-description information matches the item.

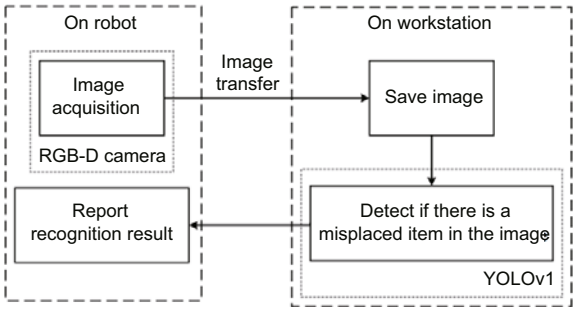

Fig. 14 Vision-based object recognition architecture

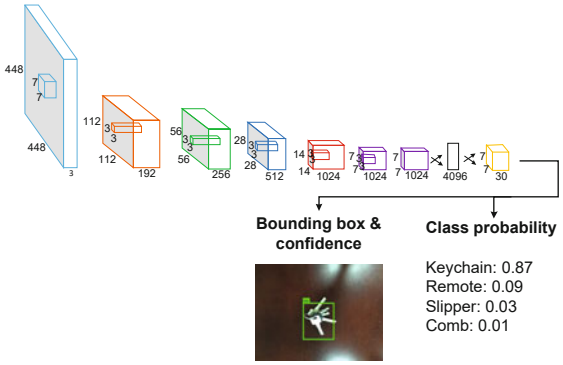

Fig. 15 Illustration of the CNN architecture (modified from Redmon et al. (2016))

We chose several kinds of misplaced items to train the CNN, including keychains, remotes, slippers, and combs. The sample training images were collected from both the ImageNet dataset (http://www.image-net.org/) and our smart home testbed captured by the robot's camera. Figs. 17 and 18 show the training examples for the keychain and the remote, respectively. The numbers of training samples in the dataset for different classes are shown in Table 2, where 70% of the sample images were chosen randomly to be used for training, 15% for

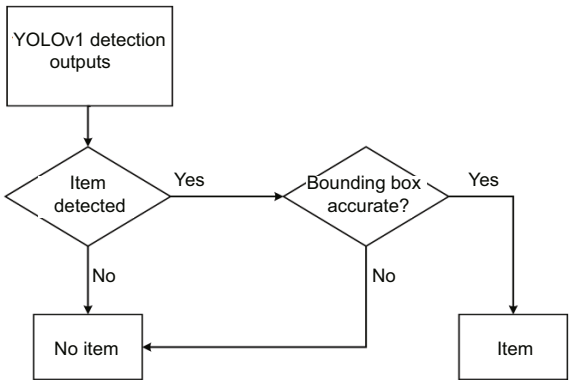

Fig. 16 Evaluation flowchart

Fig. 17 Example of the keychain training sample

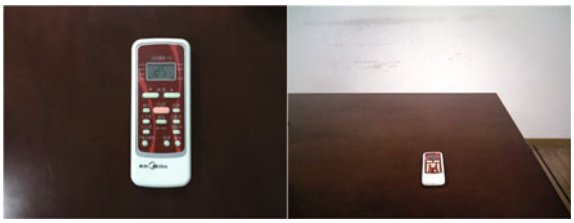

Fig. 18 Example of the remote training sample

Table 2 Sample number of the dataset for each category

Category	Number of samples			
	Keychain	Remote	Slipper	Comb
Training	867	828	638	759
Validation	186	178	137	163
Testing	186	178	137	163

validation, and 15% for testing. After 3000 epochs of training, the trained CNN correctly identified 90% of the sample images in the test set. Also, the average processing time for each image was less than 0.1 s, which means that the robot can realize real-time search with a 10 frames/s video input.

5 Experimental evaluation

5.1 Experimental procedure

The performance of our proposed robotic MIF system was evaluated in the smart home testbed as shown in Fig. 4. We chose the keychain as the misplaced item. A resident was asked to move around the mock apartment and place the keychain in several different areas, such as the bedroom, living room, and kitchen. In each scenario, the search results of our knowledge-based search strategy were compared with those of the random search strategy (Benchmark I) and the predetermined search strategy (Benchmark II), in which the robot search paths were generated by a random region transition sequence and a fixed region transition sequence, respectively. The experiments of the random search strategy and the predetermined search strategy were conducted without any complementary information, whereas the experiment of our approach was conducted with the human historical trajectory as the knowledge. The video clips of our experiments are available in Wang (2018). Fig. 19 shows that after a successful search, the robot stopped its movement in front of the keychain.

5.2 Search results

Some results of the comparative experiment are shown in Fig. 20 and Table 3. In each subfigure of Fig. 20, the magenta triangle represents the initial position of our robot, which was fixed in every search, while the green circle represents the location where our robot finds the misplaced item. The blue curve is the robot trajectory obtained from the odometer, and the red one is the trajectory of the resident. A distinct difference can be seen in the robot trajectory between the proposed approach and the benchmark methods, because our search strategy has the advantage of knowing where the misplaced item may be located through the human trajectory information.

Three different performance metrics were used to evaluate the efficiency of the proposed approach, including the total time consumption (T_{found}), the total length of the path (Len), and the total angle of rotation (Ang). The impact of the knowledge on improving the effectiveness of MIF was demonstrated in the comparative evaluation results shown in Tables 4 and 5. It can be observed that the proposed method took 49% less time to complete the search task, and an average of 47% less distance and 52% less rotation, which means that the proposed method is much more efficient in MIF than the benchmark methods.

6 Conclusions and future work

In this paper, we aimed to develop a robotic MIF system which uses the complementary information provided by a smart home environment.

Fig. 19 Robot stops when the keychain is found

Table 3 Performance of the three methods

Metric	Bedroom			Living room			Kitchen		
	I	II	P	I	II	P	I	II	P
T_{found} (s)	172.55	169.34	85.64	203.56	127.73	85.94	131.25	272.39	85.74
Len (m)	18.31	10.81	7.14	19.47	6.58	6.82	13.54	18.51	6.01
Ang (rad)	58.19	65.79	25.71	69.72	53.00	30.77	46.80	102.77	32.34

I: Benchmark I method; II: Benchmark II method; P: proposed method

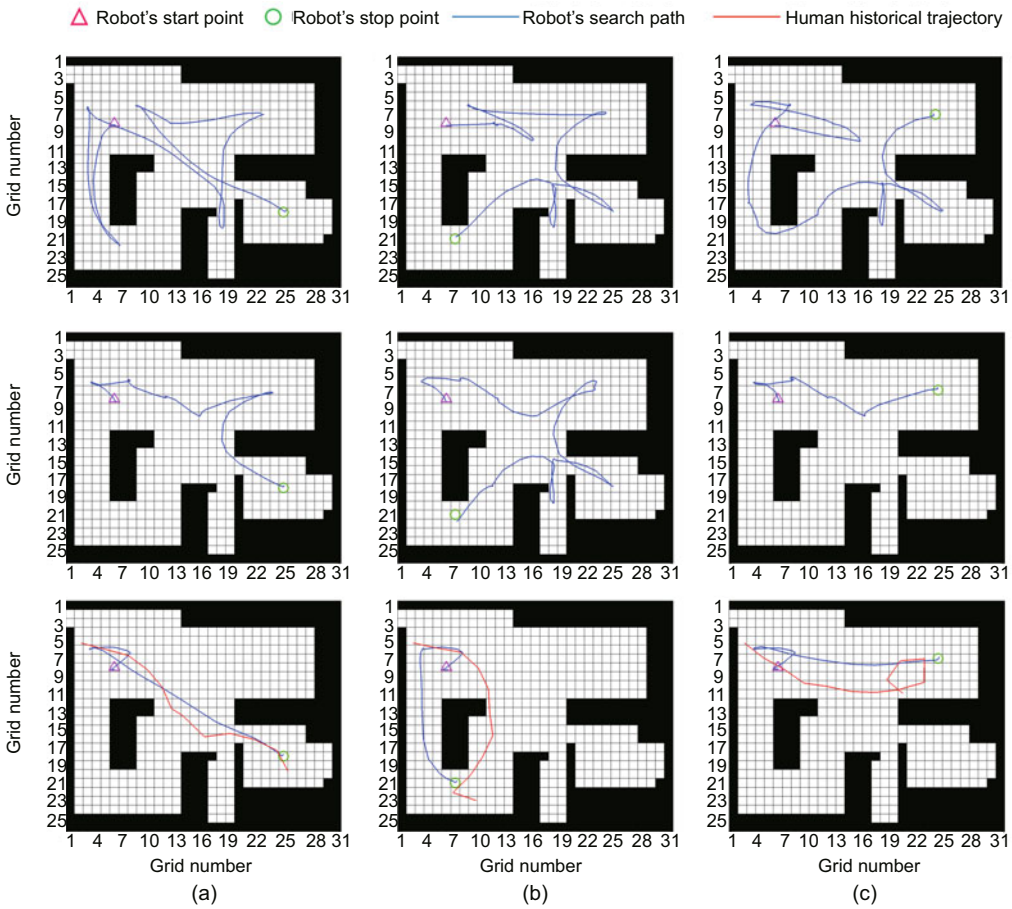

Fig. 20 Robot trajectories when the keychain was lost in the bedroom (a), kitchen (b), and living room (c). Rows 1–3 indicate the results of the Benchmark I, Benchmark II, and the proposed methods, respectively. References to color refer to the online version of this figure

Table 4 Improvement over the Benchmark I method

Area	Improvement (%)		
	T_{found}	Len	Ang
Bedroom	50.37	61.00	55.82
Living room	57.78	64.97	55.87
Kitchen	34.67	55.61	30.90
Average	47.61	60.53	47.53

Table 5 Improvement over the Benchmark II method

Area	Improvement (%)		
	T_{found}	Len	Ang
Bedroom	49.43	33.95	60.92
Living room	32.72	−3.65	41.94
Kitchen	68.52	67.53	68.53
Average	50.22	32.61	57.13

A wearable motion sensor provided the human motion model and a PIR sensor network distributed in the home provided the external observations of the rough human location, while a sensor fusion method was used to obtain the accurate human location. Given the human historical trajectory data, a modified GA has been implemented in the mobile service robot to generate a preferable search path in the indoor environment. The proposed approach has the advantage of reducing the time needed for the robotic MIF process. We conducted experiments in our smart home testbed and the effectiveness of the proposed robotic MIF system was validated.

In our future work, we will mount a manipulator on the mobile service robot that can retrieve the items found and deliver them to the user. Also, an active search mode will be integrated in the robot that allows the robot to search for items that are hidden by other objects. In addition, we will investigate replacement of the human localization

sensor network with an intelligent floor with pressure sensors to enable the localization and tracking of multiple people.

Compliance with ethics guidelines

Qi WANG, Zhen FAN, Wei-hua SHENG, Sen-lin ZHANG, and Mei-qin LIU declare that they have no conflict of interest.

References

Ahern SJ, Carter J, Wilson P, 2015. Autonomous robot retrieval system. SAI Intelligent Systems Conf, p.280-282. https://doi.org/10.1109/IntelliSys.2015.7361155

Castellanos JA, Neira J, Tardós JD, 2001. Multisensor fusion for simultaneous localization and map building. *IEEE Trans Rob Autom*, 17(6):908-914. https://doi.org/10.1109/70.976024

Choi YS, Deyle T, Chen T, et al., 2009. A list of household objects for robotic retrieval prioritized by people with ALS. IEEE Int Conf on Rehabilitation Robotics, p.510-517. https://doi.org/10.1109/ICORR.2009.5209484

Ekvall S, Kragic D, Jensfelt P, 2007. Object detection and mapping for service robot tasks. *Robotica*, 25(2):175-187. https://doi.org/10.1017/S0263574706003237

Fujii T, Ueda H, Minoh M, 2007. A looking-for-objects service in ubiquitous home. *J Nat Inst Inform Commun Technol*, 54(3):175-181.

Huynh SM, Parry D, Fong ACM, et al., 2014. Home localization system for misplaced objects. IEEE Int Conf on Consumer Electronics, p.462-463. https://doi.org/10.1109/ICCE.2014.6776087

Kollar T, Roy N, 2009. Utilizing object-object and object-scene context when planning to find things. IEEE Int Conf on Robotics and Automation, p.2168-2173. https://doi.org/10.1109/ROBOT.2009.5152831

Komatsuzaki M, Tsukada K, Siio I, et al., 2011. IteMinder: finding items in a room using passive RFID tags and an autonomous robot (poster). Proc 13th Int Conf on Ubiquitous Computing, p.599-600. https://doi.org/10.1145/2030112.2030232

Lin YC, Wei ST, Yang SA, et al., 2015. Planning on searching occluded target object with a mobile robot manipulator. IEEE Int Conf on Robotics and Automation, p.3110-3115. https://doi.org/10.1109/ICRA.2015.7139626

Meng Q, Lee MH, 2006. Design issues for assistive robotics for the elderly. *Adv Eng Inform*, 20(2):171-186. https://doi.org/10.1016/j.aei.2005.10.003

PIR, 2018. Passive Infrared Sensor. https://en.wikipedia.org/wiki/Passive_infrared_sensor

Ravi K, Bhavani KJ, Vinayak M, et al., 2016. Identification of misplaced objects. *Indian J Sci Technol*, 9(17):1-4. https://doi.org/10.17485/ijst/2016/v9i17/92991

Redmon J, Divvala S, Girshick R, et al., 2016. You only look once: unified, real-time object detection. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.779-788. https://doi.org/10.1109/CVPR.2016.91

Rogers JG, Christensen HI, 2013. Robot planning with a semantic map. IEEE Int Conf on Robotics and Automation, p.2239-2244. https://doi.org/10.1109/ICRA.2013.6630879

ROS, 2018. Robot Operating System. http://www.ros.org/

Sadhu C, Abhiram MH, Chandan B, et al., 2013. Cognitive learning enabled real time object search robot. Int Conf on Control, Automation, Robotics and Embedded System, p.1-6. https://doi.org/10.1109/CARE.2013.6733739

Schwarz D, Schwarz M, Stückler J, et al., 2014. Cosero, find my keys! Object localization and retrieval using bluetooth low energy tags. Proc 18th RoboCup Int Symp, p.195-206. https://doi.org/10.1007/978-3-319-18615-3_16

Susi M, Renaudin V, Lachapelle G, 2013. Motion mode recognition and step detection algorithms for mobile phone users. *Sensors*, 13(2):1539-1562. https://doi.org/10.3390/s130201539

Tesoriero R, Gallud JA, Lozano MD, et al., 2009. Tracking autonomous entities using RFID technology. *IEEE Trans Consum Electron*, 55(2):650-655. https://doi.org/10.1109/TCE.2009.5174435

Wang Q, 2018. Youtube Video on Robotic Misplaced Item Finding System. https://youtu.be/jEDKbr71pis

Wang Q, Zhang SL, Liu MQ, et al., 2017. Retrieval of misplaced items using a mobile robot via visual object recognition. IEEE 7th Conf on CYBER Technology in Automation, Control, and Intelligent Systems, p.1188-1193. https://doi.org/10.1109/CYBER.2017.8446158

Xu D, Huang Q, Liu HW, 2016. Object detection on robot operation system. IEEE 11th Conf on Industrial Electronics and Applications, p.1155-1159. https://doi.org/10.1109/ICIEA.2016.7603758

Xunfei, 2018. Voice Dictation. http://www.xfyun.cn/services/voicedictation