Published in final edited form as:

Cogn Syst Res. 2019 May ; 54: 258–272. doi:10.1016/j.cogsys.2018.10.032.

Robot-Enabled Support of Daily Activities in Smart Home Environments

Garrett Wilson, **Christopher Pereyda**, **Nisha Raghunath**, **Gabriel de la Cruz**, **Shivam Goel**, **Sepehr Nesaei**, **Bryan Minor**, **Maureen Schmitter-Edgecombe**, **Matthew E. Taylor**, **Diane J. Cook**

Washington State University, Pullman, WA

Abstract

Smart environments offer valuable technologies for activity monitoring and health assessment. Here, we describe an integration of robots into smart environments to provide more interactive support of individuals with functional limitations. RAS, our Robot Activity Support system, partners smart environment sensing, object detection and mapping, and robot interaction to detect and assist with activity errors that may occur in everyday settings. We describe the components of the RAS system and demonstrate its use in a smart home testbed. To evaluate the usability of RAS, we also collected and analyzed feedback from participants who received assistance from RAS in a smart home setting as they performed routine activities.

Keywords

smart homes; activity learning; robot assistance

1. Introduction

The world's population is aging – the estimated number of individuals age 85+ is expected to triple by 2050 [1]. Currently, an estimated 50% of adults age 85+ need assistance with everyday activities and the annual US cost for this assistance is nearly $2 trillion [2]. In the case of dementia, cognitively-complex functional activities such as using the phone, preparing meals, taking medications, and managing finances are affected early in the course of the disease [3]–[5]. Moreover, functional impairment in older adults, or the inability to independently perform these tasks, has been associated with increased health care use, and placement in long-term care facilities [6], [7], number of days in the hospital [8], poorer quality of life [9], [10], morbidity, and mortality [11]. According to the National Aging in Place Council [12], upwards of 90% of older adults prefer to age in place as opposed to moving into a nursing home. Alternative forms of health care are therefore needed to preserve older adults' independence and quality of life. Technologies that automatically

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

assist with activity of daily living may relieve some of the strain on the health care system as well as caregivers, allowing individuals to remain functionally independent and age in place.

Previous research suggests that smart home technologies, powered by machine learning and automated reasoning, can provide insights into a person's health status [13]–[18]. Information from smart homes can also be harnessed to create reminder systems [19] as well as automate control of home devices [20]. What smart homes do not normally bring is a tangible, mobile avatar that partners with the smart home to proactively provide assistance.

In this paper we introduce RAS, a multi-agent robot system that is designed to provide in-home activity support for older adults and others that need assistance to independently perform basic and instrumental activities of daily living (ADLs). RAS represents a collaboration between a smart home and a mobile robot. In this partnership, the smart home tracks activities of daily living and determines when the resident needs assistance. The robot represents an interactive, assistance agent that provides the assistance in the moment, as the need for help is detected. We describe the vision for RAS as well as the details of its design and implementation. We also demonstrate RAS capabilities with a participant performing activities in a smart home testbed.

2. Related work

Many elder-care robots are already being designed or are in production, representing a wide assortment of appearances and purposes. One such purpose is companionship, exemplified by the robot Paro [21]. Paro resembles a baby seal and mimics animal-assisted therapy to elicit feelings of joy, happiness, and relaxation from its owner without facing the potential dangers of a real animal such as bites. Pepper is another social companion robot that perceives its owner's emotions and adapts its behavior accordingly [22]. Because Pepper can move, talk, and display information on a tablet, the robot can also talk with the user about the news or weather.

A second role that assistive robots play is that of coach. For example, robots can motivate older adults to engage in physical exercise. The robot coach can demonstrate exercises that users need to perform for fitness or for physical therapy [23]. Similarly, a robot coach can promote strong social interaction strategies for individuals needing autism behavior therapy or stroke rehabilitation [24]. In the case of the robot nurse Pearl, the robot can guide users to where they need to go in order to make important appointments [25].

A third role for robots in elder care is physical assistance. A well-known example is Asimo. Designed by Honda, Asimo helps older adults who have physical limitations. Because Asimo [26] is mobile and it can fetch food and turn devices on or off in the home, the robot can provide physical assistance to individuals. Robear, a nursing robot, can pick patients up and transfer them between bed and wheelchair [27].

Some assistance for activity prompting has been provided in past approaches through a speaker or computer, but studies have shown that users are more responsive to a physical robotic coach than a virtual one [23]. Research also indicates that the way in which the robot approaches a human and leads a human around a space is important. For example, Hoffman et al. [28] found that robot movements and gestures are socially expressive and will affect the engagement of the human into the interaction. Similarly, Kupferberg et al. [29] confirmed that human-like velocity in robot movement facilities perception of a humanoid robot as a true interaction partner. Destephe et al. [30] and Bisio et al. [31] confirmed that humans better understand robot intent (such as leading them to where a key object may be found) if the robot moves in ways that mimic biological movement patterns.

Our RAS system also introduces a robot that plays the role of a coach-style demonstrator as well as physical assistant. Two systems that share similar goals with this work have been introduced by Begum et al. [32] and by Bovbel at al. [33]. The robotic activity support system developed by Begum et al. [32] uses a camera system to track participants as they perform two activities in a laboratory setting. The robot system identifies errors consisting of skipped steps. When an error is detected, an operator tele-operates the robot to where the human is at and plays a video recording of the missed step. Bovbel et al. [33] also designed a robot system that aids with activities, specifically kitchen-based cooking activities. The location of the human is provided to the robot, which searches for a path to reach the human and guide them to the kitchen. Once in the kitchen, the robot points to items at preprogrammed locations that can be used in a recipe that the human selects.

Like these previous approaches, the RAS robot coaches users on needed activity steps so that residents can perform their daily activities accurately and completely. RAS extends previous work to detect when there is a need for such assistance based on sensor-driven detection of activity errors. Unlike previous approaches, RAS searches for and locates the humans, automatically provides video reminders, and leads them to the detected objects that are required for the skipped activity steps. Furthermore, RAS provides physical assistance by showing residents where key objects are in the home that they need for daily activities. As we will describe in the following section, a physical robot can uniquely extend the assistance of a smart home environment by approaching the resident when a need for help is detected and by physically guiding the resident to find objects needed for daily tasks.

3. Robot Activity Support

Memory has a large impact on everyday function, particularly as cognition-affecting diseases progress. RAS represents a technology that aids with everyday function by acting as a cognitive prosthesis. Figure 1 illustrates the Robot Activity Support (RAS) system components. As shown in the figure, sensors are embedded inside a smart home environment. Based on collected sensor data, models are built of typical activity performance and provide a basis for detecting errors in activity performance. Activities can contain a variety of errors. Previous work [34] categorizes activity errors as errors of omission (a needed activity step is not performed), substitution (an unusual object is used), irrelevant action (an action unrelated to the activity is performed), and inefficient action (an action is performed that slows down the task). RAS currently detects omission errors and substitution errors. In some contexts, inefficient actions or irrelevant actions may be detected if they cause an activity step to not occur at the expected point in the sequence.

When an error is detected, a message is sent to the physical robot to provide assistance. The robot finds and approaches the resident and asks if they would like help with the activity. if they request help, the user has three options. First, they can indicate through a tablet interface that they would like to see a video of the activity step they should be performing next. Second, they can indicate that they want to see a video of the entire activity. Third, they can alternatively ask the robot to show them where the object is that they need for the current activity step. The RAS robot uses the onboard tablet to play the requested videos. If, however, the resident needs help finding the object, the robot will retrieve its learned map of the home and object locations and use the information to lead the resident to the object.

For RAS to be capable of performing all of these tasks the system requires a variety of functions. These include a navigation module to create a map of the house, localize itself within that map, and move to specified locations within the house; an object detection module to visually recognize objects (as well as the human resident) coupled with a depth sensor that determines the 3D object location relative to the map for storage and retrieval; a user interface module that can prompt the resident if he or she needs help as well as offer relevant activity information; an error detection module to recognize when the resident has made an error and may require assistance; and a process manager to control all of the modules. All of the RAS modules are implemented in the ROS robot operating system [35]. Figure 2 diagrams these key components, the locations of the components and needed data on various devices, and the flow of communication between the components. The components are described in detail in the following sections.

As an example of a RAS interaction, consider the task of eating breakfast. RAS must identify that the resident is eating breakfast and that the resident typically eats cereal. RAS may then assist the resident by suggesting that it is time for breakfast (if the resident does not initiate the task at the normal time), showing the entire breakfast routine by video if needed, showing a video of the next step if the resident is stuck mid-activity (e.g., cannot remember where to find the bowl), and leading the resident to the needed object such as a bowl if they cannot find it.

3.1. Robot Platform

The physical design of the robot platform must allow the robot to interact naturally with the user (i.e., voice and/or video), allow the robot to understand what steps the user has and has not yet accomplished (i.e., use sensors on the robot and/or in the home to detect the environment and the resident actions), and provide assistance (i.e., show a video or find an object). The robot must be able to move on level hard floors as well as carpet, have an onboard camera, and carry a tablet at chest-level that acts as the main user interface for RAS.

Our robot is based on a design by Eaton, et al. [36] with modifications in its mechanical design. Specifically, we increased the height of the robot by one foot and added a 4 ft mast of lightweight extruded aluminum that further raises the height to 5.5 ft. We also changed the pan-tilt mount design for positioning the 3D camera by adding an extra small mast in the horizontal direction. This better accommodates the Arduino board and actuators which automate the pan-tilt mount. All cables are routed through the mast to the Turtlebot base.

Some of the major mechanical challenges with the prior work were weight limitations (with a maximum payload of 5 kilograms) and instability of the moving platform on rough surfaces. To tackle these problems, a transformation robot was built on a Turtlebot 3 platform which is a two-wheeled differential drive type platform with a maximum payload of 30 kg. To improve the quality of maneuvers and avoid wobbling, we added 3D printed wheels, “Omni wheels”, at the back of the platform. These wheels allow the robot to move in all directions. We utilize an Intel Joule computer and OpenCR controller board for robot control and a Nvidia Jetson computer for object detection and object location storage. The robot base employs 2D LiDAR for navigation. To facilitate object/human detection, we installed an Orbbec Astra RGBD camera on the top of the robot. Finally, we added an Android tablet for user interaction. The robot platform is shown in Figure 3.

3.2. Management Module

To connect the RAS components, we implemented a manager module. The manager is implemented within ROS State Machine (SMACH) [37], a task-level architecture for rapidly creating complex robot behavior. Specifically, we constructed two state machines: a “Go to Person” SMACH and a “Go to Object” SMACH, as shown in Figure 4. These state machines are reflective of the fact that activity errors are currently detected based on lack of use of the correct activity object at the correct point in the sequence, as described at the beginning of Section 3. Within a SMACH, a “state” is a local execution state or a single task that the system performs. In Figure 4, each oval represents a state executed by the system.

Once the “Go to Person” SMACH is triggered by the manager, the “FindPerson” state is initiated. This state first queries the database for the location of the human and a corresponding timestamp when the human was detected. If the timestamp has expired (i.e., it is at least 10 seconds old), the system looks for the human by rotating the camera to scan the space. Upon successfully locating the human, the state ends which gives rise to a new state, “GotoXY”. If the human is not located then the robot moves to a new location (in the new state “GotoNewBase”) and repeats the search. When found, the human’s location is sent (via “GotoXY”) to the navigation system. A similar SMACH controls the robot in search of objects that are used in daily activities.

3.3. Navigation Module

States “GotoXY” and “GotoNewBase” are handled by the navigation module. The navigational map is built using Google’s Cartographer system [38]. The Cartographer’s simultaneous localization and mapping (SLAM) [39] algorithm constructs a map of the smart home (as shown in Figure 5) and dynamically determines the robot’s location on the map. The map is available to all components using ROS’s publish/subscribe service. Desired XY map coordinates are given to the ROS movebase to move the robot to the goal location as well as to detect and avoid possible collisions. In order to create an initial accurate map of the space, a human driver manually directs the robot around the environment. The robot moves while assisting the resident as well, which provides an opportunity to dynamically update the map.

Navigational obstacles are detected in the environment by the robot's LiDAR. We analyze both the raw data sent by the sensor and data that is filtered through Cartographer. The combined information is used to generate an effective cost map, which indicates points in the space that will cause collisions with our system (as shown in Figure 5). These collision points are then expanded to create a safe area around detected obstacles that prevents "close call" collisions between the RAS robot and the physical environment.

To account for objects which exist primarily outside of the LiDAR range, we create navigation paths that leave a maximum amount of clearance on either side of the robot. For an indoor environment, this offers a practical strategy because in most situations the robot will detect opposing walls at all times. Using SLAM, the cost map, and this path-building strategy, the robot can navigate to the resident and to the needed objects throughout most smart environment spaces. In our smart home testbed the robot was able to navigate to approximately 90% of the points in the space. The remaining 10% of the home represent regions that the robot could not reach with sufficient clearance to nearby obstacles.

3.4. Object Detection Module

RAS requires an object detection module to visually find objects of interest as well as find the human resident. RAS needs to keep track of where objects are that are typically used for daily activities so that if the resident forgets where an object was left, the robot can lead the person to the object. We implement object detection using an RGB-D camera. Here, RGB refers to the red, green, and blue channels of a color image and D refers to an additional depth channel, where depth reflects the distance to the nearest object at each pixel from the RGB image.

The RGB-D camera is positioned on an Arduino-controlled pan-tilt mount that is integrated into the ROS system. In addition to panning and tilting via two servos, the Arduino reports to ROS the servo angles. These pan/tilt values enable calculation of where the camera is pointed with respect to the robot. RAS can thus scan the environment to find objects with respect to the robot's map through a coordinate transformation. RAS currently requires that the objects be placed in plain view for the robot, such as sitting on a table or hanging in a closet with the door open.

From the RGB image, the object detection algorithm generates a bounding box around each object of a given class (e.g., each medicine container) within the image. Once the bounding box is created, 3D positions of each bounding box are calculated based on the calibrated depth channel. Next, the 3D position relative to the camera is converted into a position relative to the navigation module's map. This conversion is performed by applying a coordinate transform to the original 3D location. Finally, we update the last-seen location of the object and save it to a database that RAS can later query when the object is needed.

A common technique for performing object detection is by using a convolutional neural network (CNN). In this process, a collection of boxes of different sizes are overlaid atop an image, each of which provides a data point for training the network. The network is trained to predict a class (object label) for each box as well as an amount and direction to shift the box to accurately cover the ground truth object bounding box. During training, an objective function is used that balances minimizing error between incorrect class labels and minimizing error due to incorrect bounding box positions [40]. Utilizing a complex objective function results in many parameters that need to be refined for object detection. These include the structure of the network and the objective function. They also include choices for bounding box numbers, sizes, and positions.

We consider three types of CNN-based object detection strategies [40]. These include single shot multibox detectors (SSD), regions with convolutional neural networks (R-CNN), and region-based fully convolutional network (R-FCN). SSD makes a single pass through the CNN to predict both classes and bounding boxes. In contrast, R- CNN performs two steps. In the first step, a portion of a feature extractor network processes the image and proposes boxes that may contain the objects. After cropping the image to contain only the boxes, the updated features are processed by the rest of the network in the second step to classify the objects. R-FCN performs similarly to R-CNN except the image is cropped at the penultimate layer of the network rather than in the middle of the feature extraction network.

With any of these network choices, object detection CNNs can be trained using a combination of object images from the smart home and images of similar objects from an online database. Each image is preprocessed by manually drawing a bounding box around the object of interest and giving it a label. Once trained, the object detection algorithm runs in real time from video captured on the RAS robot. The algorithm processes approximately 10 frames per second, which allows for image processing at the normal speed of movement for the robot. The training and use process is illustrated in Figure 6 based on an SSD detector [41]. During RAS operation, the point cloud generated from the camera's depth sensor indicates the 3D position of the detected object relative to the camera. ROS's tf2 library transforms the relative position to a position that can be stored in an object location database as the last-seen location of this object.

The last-seen location of each object can then be integrated into the error detection and recovery mechanisms. When a task error is detected, the first step is to navigate the robot from its base location to the person. This requires knowledge of the human's location. One of the "objects" that the object recognition system detects is the human, so the last-seen location of the human is queried from the database along with the timestamp. If the timestamp has not expired, then the navigation goal is set to this location, and the robot approaches the human to display several options for assistance. If the timestamp has expired, then the robot will navigate to a known location with high visibility (such as the entrance to a hallway) and rotate the camera to find the human. In our experiments, this method of finding the human was largely successful due to the majority of the apartment being visible from a couple of key locations.

If the detected error involves an object recognized by the object recognition system, one of the options for error recovery is to lead the person to that object. At this point, the robot can again query the database to retrieve the last- seen location of that object. This location is then set as the robot's navigation goal as the robot says "follow me" via its speaker. The robot then autonomously drives to the location and rotates to face the object. Upon arrival, it says "here you go" to indicate the object should be in front of the robot. Thus, through object recognition with the camera and maintaining a list of last-seen locations, the robot is able to both navigate to the human and also navigate to objects for error recovery.

3.5. User Interface Module

RAS's onboard tablet provides an interface between the human and the robot. The interface is a website that is displayed on the tablet. Connections between the web browser, the robot, and the retrieved videos are shown in Figure 7. The ROS package rosbridge provides a web socket that is accessible via Javascript. The Javascript library facilitates creating and accessing ROS publish/subscribe notifications, services, and actions through the webpage.

Studies indicate that older adults prefer less anthropomorphic designs for robot assistants [42]. To provide interaction with the smart home resident that is not overly humanoid, the RAS interface module includes several artist-rendered faces that indicate the "emotion" of the robot. As shown in Figure 8, these expressions can be either neutral, questioning, or pleased. The neutral face is the robot's resting face and is supplied when the robot is observing user tasks in the home. If an activity error is detected, then a questioning expression is displayed along with a text query and automatically-voiced query whether the user needs activity help. If the user does request videobased help, then videos of either the current activity step or the entire activity are pulled from a server and displayed on the web interface. If the user requests RAS to take them to the needed object, then the automated audio prompt "Okay, follow me" is played before the robot leads the user to the object. Once the error is corrected and the activity step is correctly performed (or the user selects "I did it!" on the interface), RAS displays a pleased expression.

3.6. Smart Home-Based Error Detection Module

Smart homes and other smart environments provide a timely opportunity for automated health monitoring and intervention. In these environments, sensor data is collected while residents perform daily routines. Unlike many other forms of behavioral data collection, smart home-based data collection can be performed without making any changes to a resident's routine activities. This provides a more ecologically valid mechanism for functional health observing, assessing, and intervening. Recent advances have transformed smart homes from experimental prototypes to operational assistive environments [43], [44]. One such smart home design, CASAS, partners with RAS to provide activity tracking and intervention support.

In a smart home environment, the home acts as an intelligent agent that perceives the state of the physical environment and its residents using sensors, reasons about this state using artificial intelligence t echniques, and then takes actions to achieve specified goals [45]. During perception, control flows from the physical components through the middleware to the software applications. When taking an action, control moves back from the application layer to the physical components that automate the action. As a class of physical components, sensors in the home generate readings while residents perform daily activities. The sensor readings are collected by a computer network and sent to a middleware manager that tags each reading with a time stamp and an identifier for the sensor that generated the reading. The sensor data is then collected in a relational database that an intelligent agent can use to generate useful knowledge such as patterns, predictions, and trends. If an activity error is detected, then a decision will be sent through the middleware to the robot at the physical layer and the robot executes the selected action (e.g., interact with the user, play a video, or find a needed object).

The CASAS physical layer contains hardware components including sensors and actuators. The software architecture utilizes a wireless network to control communication between physical and software components. The middleware, which resides on a Raspberry Pi, contains a publish/subscribe manager. Every component in CASAS communicates to the manager by means of a XMPP bridge. Examples of such bridges are the Zigbee / wireless bridge, the Scribe bridge which archives sensor data, inferred patterns, and selected decisions in a secure database, and bridges for each component in the application layer.

When RAS is used “in the wild” it will rely on activity recognition to determine a smart home resident’s current activity and track activity steps to detect possible errors. Many diverse approaches have been explored for activity recognition in smart homes [46]–[51]. Some of these techniques have demonstrated the ability to perform real-time activity recognition in home settings and some work has been explored to detect errors within the progression of simple and complex activities [52]. In the current RAS approach, a directed acyclic graph (DAG) is constructed for each activity showing the order in which activity steps should be performed. In near real-time, when steps are detected in a non-DAG-supported order, then the CASAS system initiates an error state and communicates the error step to the management module. The error detection module will wait for a message from the management module that the error state is resolved before it continues monitoring the remaining activity steps. In this first RAS study, the activities to monitor are given to the module at the start of the experiment.

Although a variety of sensors can be employed in a smart home environment, for our initial RAS implementation we use Estimotes [53]. Estimote units contain accelerometers and temperature sensors embedded within stickers that can be attached to key items around a house. Activity steps can then be detected based on object usage. To collect Estimote data into the CASAS infrastructure, we designed an app that receives time-stamped Estimote data via Bluetooth and sends corresponding start time and stop time information to the middleware manager each time an object is used.

4. Evaluation of RAS activity assistance

To demonstrate the ability of RAS to perform activity assistance, we constructed several scenarios in which smart home residents performed activities in a smart apartment testbed. Twenty-six participants were recruited to perform the activities, interact with the robots, and provide feedback on the experience. While each participant performed the activities and interacted with the robot, two experimenters observed from a remote location and a third experimenter stayed on site to assist with the robot if needed.

Participants completed three scenarios representative of activities older adults might perform: preparing to take the dog for a walk, taking medicine with food and water, and watering the household plants. Figure 9 shows screenshots from the medicine scenario. The participants completed each activity once without errors and an additional three times with a different omission error. A summary of the activity steps and error conditions is given in Table 1.

When RAS detected an error, the robot detected and approached the human to offer help. The three assistance prompts were:

1. Watch a video of the entire activity.
2. Watch a video of the next step in the activity sequence.
3. Lead the participant to the necessary object for the next activity step.

Suggested responses to these options were counterbalanced across participants and activities to evaluate a variety of assistance conditions.

4.1. Evaluation of object detection

The ability of RAS to provide activity support depends in large part on its ability to detect both humans (in order to find and approach the smart home resident) and to detect objects used in daily activities (to keep track of their location and lead the resident to the object when needed). To train RAS for our example activities, we collected 2,433 sample pictures of humans and objects in the smart home testbed. Because the object detection model needed to be able to generalize to multiple smart home residents, we also downloaded human images from the Microsoft COCO dataset [54]. Using all 64,115 available online images would create a significant class distribution imbalance. To address this issue, we randomly selected 20,000 of the COCO human images and oversampled the manually collected images by duplicating randomly-selected images from each category until the class distribution was fairly uniform, as shown in Table 2. Prior to sampling, the images were split by class: 80% of the original images were used for sampling and training, 20% of the original images were held out for testing.

Before integrating object detection into RAS, we evaluated alternative convolutional neural networks on the object classes that we use for our current activity scenario. Because the networks output a bounding box for the detected object as well as an object label, a performance measure needs to be used that assesses the “correctness” of the area bounded by the box. Here we use 0.5 IoU. IoU represents the “intersection over union”, or the intersection of the predicted bounding box and the ground truth bounding box divided by the union of the two. For the network-generated bounding box to be “correct”, the intersection must be at least 50% of the union, or 0.5 IoU. Based on this measure, Table 2 summarizes object detection precision for the R-CNN, R-FCN, and SSD networks. As the table shows, all networks performed very well on the inanimate objects. Detecting humans is much more challenging due to the variation in human features and future work can focus on designing methods to improve detection of this category [55]. Although they perform comparably, we use SSD on the robot because of its smaller memory and computational resource requirements.

To further evaluate the performance of the object detectors, in Figure 10 we graph the precision-recall curves by class for the faster R-CNN, R-FCN, and SSD networks. These results are consistent with the findings in Table 2 in that the curves indicate fewer false positives and false negatives for the inanimate objects and more difficulty in detecting the human within the environment.

4.2. Evaluation of robot activity support

The long-term goal of RAS is to partner smart environments and robots to provide activity support for individuals who need help staying functionally independent. With this in mind, evaluation of RAS needs to consider not only the accuracy of the components and overall system but also the usability of the system for its intended purpose. For each detected error, RAS found and approached the human, then either displayed the selected video or led the participant to the required object. Our experimental results are based on a pilot test of RAS involving a sample of 26 healthy younger adults (mean age 24 years old, 42% female). As represented by a mean score of 6.09 out of 7 on a technology comfort questionnaire, participants self-reported a high degree of daily use and comfort with technologies.

Survey data was collected to assess RAS usability with the Post-Study System Usability Questionnaire (PSSUQ; Likert ratings: 1 = strongly agree, 7 = strongly disagree). As seen in Table 3, mean ratings of the overall RAS system along with the three subscales ratings (i.e., 4.37 – 4.83) fell within the neutral range leaning more negative (disagree). Of note, there was a lot of variability in participant responses to the PSSUQ usability items, suggesting significant individual differences, with responses spanning the entire range of strongly agree to strongly disagree. In contrast, the user satisfaction questionnaire (i.e., QUIS) Screen and Terminology subscale scores showed more universally favorable reviews of the RAS tablet interface (7.96 and 7.74 out of a possible 9.00, respectively).

To determine prompt type preference (guide to object, next step video and full video) for four separate purposes (see Table 4), participants were asked to rate each presented prompt type on a 1 to 5 Likert scale, with 1 representing least easy/helpful or most confusing and 5 representing easiest/more helpful or least confusing (see Table 4). Repeated measure analyses of variance with prompt type as the within subject variable were conducted and followed by Least Significant Differences (LSD) post hoc tests. For all four prompt purpose questions, significant differences emerged across prompt types, *Fs* > 213.00, *ps* < .001. Follow-up tests revealed that compared to the full video, participants rated the next step video as significantly easier to imitate, *F*(1, 20) = 4.96, *p* = .04, and more helpful if unable to recall the next step, *F*(1, 21) = 6.02, *p* = .02. Follow-up tests also revealed that participants rated the next step video as likely to be both more helpful for someone with mild cognitive impairment (MCI) and less confusing to someone with MCI than both guidance to object, Fs > 5.01, *ps* < .05, and the full video, *F*s > 5.86, *p* < .05. As seen in Table 5, this was further supported by participant choices of the most liked and helpful cues, with the next step video being chosen by 60% as the most liked prompt and by 52% as the most helpful prompt. Similarly, the full video was chosen by 80% as the least effective prompt.

Finally, correlational analyses revealed no relationship between questionnaire scores (i.e., PSSUQ and QUIS) and the variables of age (*rs* between −.09 and .17) and comfort with technology (*rs* between −.33 and −.24). This suggests that the satisfaction with and usability ratings of the RAS system did not appear to differ based on participant age or technology comfort levels. Qualitative feedback revealed one consistent area for improvement: the robot's speed, both in terms of detecting errors and moving towards the participant.

5. Discussion and Conclusions

A smart environment can use heterogeneous devices to not only monitor and assess resident activities but also intervene to support daily activities. In this project we introduce RAS, a robot activity support system that links smart environment technologies with physical robots to provide aid in completing daily activities. As highlighted in this paper, fully implementing such a system requires synergy between many components including map building, object detection, navigation, user interface, and activity error detection.

Results with participants in a smart home testbed indicate that such robot support can be usable in practical settings. The long-term benefits of providing such physical assistance for older adults ripples across multiple domains. Financial strain on the healthcare system may decrease, allowing funds to be allocated toward tailored care for individuals that cannot support themselves even with this type of robotic assistance. caregiver responsibilities may be shared with a robotic aid who can support activities around the clock. Most importantly, aiding functional health may preserve an older adult's independence and quality of life.

The RAS project represents an early step toward automated care in everyday settings. Overall impressions of the RAS tablet interface were universally favorable. However, ratings of system usability were more neutral and highly variable. Questionnaire ratings were not systematically related to age or comfort with technology. compared to the next step video and guidance to object, the full script video was rated by study participants to be less helpful and more confusing to someone with Mci. These experiments highlighted directions for future research that will solidify the impact RAs can have on independent living. For example, there is a need to assess RAs usability specifically with older adults and individuals experiencing cognitive limitations. Future work can also involve these participants in participatory design of the RAS interface and human-robot interaction that is best suited to this target group.

one of the most difficult components of RAs is object detection, specifically detecting humans in a home environment. During development of the system, RAS experienced many false positive detected humans as lighting conditions changed in the home. Future research may improve this through incorporating additional training examples and tracking humans in the environment to not lose sight of them. Additionally, the current RAS system assumes a static environment in which objects are not moved outside of the robot field of view. In future work, RAS can dynamically update the map. One approach would be for RAS to occasionally scan the environment to ensure objects are in the stored locations. When they are not, RAS can systematically search the environment for the new object location.

Additionally, open-ended comments from participants included the suggestion to make the robot response to detected errors faster and more fluid. one source of delay is the need to remotely communicate between sensors, database, robot interface, and navigation system. if the delay between these modules requires more than a few seconds then the offer of assistance loses timing accuracy and is more likely to be ignored or no longer be accurate by the time assistance is offered. Because the robot velocity is approximately 3/4 the human's walking speed, this may need to be improved as well to make assistance more timely. Further work is also needed to make RAS sensitive to a greater variety of activity error types as well as occurrence of multiple simultaneous activity errors. Existing research in monitoring the progression of automated plans [59] can be incorporated to facilitate this direction of research.

Although the current RAS design analyzes Estimote-based object use information to track activities and detect errors, error detection could be improved by including additional sensors. A next step for RAS is to include information about ambient sensors, such as motion, door, light, and temperature sensors, into the activity recognition and error detection. These sensors can also partner with the object detection module to track the resident's location in the home.

A long-term goal for this research is to test RAS in home environments over days or weeks in realistic settings, rather than relying on scripted activities. Providing continual support in home settings will not only require efficient response times and robust components but a continuous power source as well. Although the types of activities that are monitored may be limited initially, this type of RAS assistance will also require a more robust approach to activity error detection that can detect and track alternative approaches to completing a task. For practical settings, robot assistants need to dock and charge themselves. Future extensions of RAS may also include the ability to retrieve needed items as well as keep track of their locations to assist individuals with physical limitations. Evaluating RAS in homes of older adults will determine the success of such robotic systems to truly partner with smart environments in a way that assists individuals with their daily routines, improving their quality of life and preserving their functional independence.

Acknowledgements

The authors would like to thank Reanne Cunningham and Luke Etherton for their assistance with the initial design of the robot platform and study parameters. We would also like to thank Brian Thomas for his contribution to the website interface. This research is supported in part by the National Institutes of Health under Grant R25EB024327 and by the National Science Foundation under Grant 1734558.

References

[1]. Alzheimer's Association, "Alzheimer's disease: Facts and figures," Alzheimer's Dement, vol. 13, no. 4, pp. 325–373, 2017.

[2]. Ernst RL and Hay JW, "The US economic and social costs of Alzheimer's disease revisited," Am. J. Public Health, vol. 84, pp. 1261–1264, 1994. [PubMed: 8059882]

[3]. Farias S, Mungas D, Reed B, Harvey D, Cahn-Weiner D, and DeCarli C, "MCI is associated with deficits in everyday functioning," Alzheimer Dis. Assoc. Disord, vol. 20, pp. 217–223, 2006. [PubMed: 17132965]

[4]. Perneczky R, Sorg C, Hartmann J, Tosic N, Grimmer T, Heitele S, and Kurz A, "Impairment of activities of daily living requiring memory or complex reasoning as part of the MCI syndrome," Int. J. Geriatr. Psychiatry, vol. 21, pp. 158–162, 2006. [PubMed: 16416470]

[5]. Schmitter-Edgecombe M, Woo E, and Greeley D, "Characterizing multiple memory deficits and their relation to everyday functioning in individuals with mild cognitive impairment," Neuropsychology, vol. 23, pp. 168–177, 2009. [PubMed: 19254090]

[6]. Desai A, Grossberg G, and Sheth D, "Activities of daily living in patients with dementia: Clinical relevance, methods of assessment and effects of treatment," CNSDrugs, vol. 18, pp. 853–875, 2004.

[7]. Marson D and Hebert K, "Functional assessment," in Geriatric Neuropsychology Assessment and Intervention, 2006, pp. 158–189.

[8]. Sonn U, Grimbyand G, and Svanborg A, "Activities of daily living studied longitudinally between 70 and 76 years of age," Disabil. Rehabil, vol. 18, pp. 91–100, 1996. [PubMed: 8869511]

[9]. Peres K, Chrysostome V, Fabrigoule C, Orgogozo J, Dartigues J, and Barberger-Gateau P, "Restriction in complex activities of daily living in MCI," Neurology, vol. 67, pp. 461–466, 2006. [PubMed: 16894108]

[10]. Barberger-Gateau P, Dartigues J, and Letenneur L, "Four instrumental activities of daily living score as a predictor of one-year incident dementia," Age Ageing, vol. 22, pp. 457–463, 1993. [PubMed: 8310892]

[11]. Nourhashemi F, Andrieu S, Gillette-Guyonnet S, Vellas B, Albarede J, and Grandjean H, "Instrumental activities of daily living as a potential marker of frailty: a study of 7364 community-dwelling elderly women (the EPIDOS study)," J. Gerontechnology, vol. 56A, pp. M448–M453, 2001.

[12]. Lerner M, "New online service targets aging-in-place residents," The Washington Post, 2015 [Online]. Available: https://www.washingtonpost.com/news/where-we-live/wp/2016/01/19/new-online-services-targets-aging-m-place-residents/?noredirect=on&utm_term=.f31adc410766.

[13]. Cook DJ, Dawadi P, and Schmitter-Edgecombe M, "Analyzing activity behavior and movement in a naturalistic environment using smart home techniques," IEEE J. Biomed. Heal. Informatics, vol. 19, no. 6, pp. 1882–1892, 2015.

[14]. Dawadi P, Cook DJ, and Schmitter-Edgecombe M, "Automated clinical assessment from smart home-based behavior *data,"* IEEE J. Biomed. Heal. Informatics, 2016.

[15]. Dawadi P, Cook DJ, and Schmitter-Edgecombe M, "Automated cognitive health assessment using smart home *monitoring of complex tasks,"* IEEE Trans. Syst. Man, Cybern. Part B, 2012.

[16]. Dawadi P, Cook DJ, Schmitter-Edgecombe M, and Parsey C, "Automated assessment of cognitive health using smart home technologies," Technol. Heal. Care, vol. 21, no. 4, pp. 323–343, 2013.

[17]. Dawadi P, Cook D, and Schmitter-Edgecombe M, "Modeling patterns of activities using activity curves," Pervasive Mob. Comput, 2015.

[18]. Dawadi P, Cook DJ, and Schmitter-Edgecombe M, "Longitudinal functional assessment of older adults using smart home sensor data," IEEE J. Biomed. Heal. Informatics, 2015.

[19]. Minor B, Doppa JR, and Cook DJ, "Learning activity predictors from sensor data: Algorithms, evaluation, and applications," IEEE Trans. Knowl. Data Eng, 2017.

[20]. Mennicken S, Vermeulen J, and Huang EM, "From today's augmented houses to tomorrow's smart homes: new directions for home automation research," in ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2014, pp. 105–115.

[21]. Wada K, Shibata T, Asada T, and Musha T, "Robot therapy for prevention of dementia at home," J. Robot. Mechatronics, vol. 19, no. 6, p. 691, 2007.

[22]. Qureshi AH, Nakamura Y, Yoshikawa Y, and Ishiguro H, "Robot gains social intelligence through multimodal deep reinforcement learning," in IEEE-RAS International Conference on Humanoid Robots, 2016, pp. 745–751.

[23]. Fasola J and Mataric MJ, "A socially assistive robot exercise coach for the elderly," J. Human-RobotInteract, vol. 2, no. 2, pp. 3–32, 2013.

[24]. Mataric MJ, "Socially assistive robotics: Human augmentation versus automation," Sci. Robot, vol. 2, pp. 1–3, 2017.

[25]. Pineau J, Montemerlo M, Pollack M, Roy N, and Thrun S, "Towards Robotic Assistants in Nursing Homes: Challenges and Results," Rob. Auton. Syst, vol. 42, no. 3–4, 2003.

[26]. Honda, “Technical information,” Asimo, 2007 [Online]. Available: http://asimo.honda.com/downloads/pdf/asimo-technical-information.pdf.

[27]. Riken, “The strong robot with the gentle touch,” News and Media, 2015 [Online]. Available: http://www.riken.jp/en/pr/press/2015/20150223_2/.

[28]. Hoffman G and Ju W, “Designing robots with movement in mind,” J. Human-Robot Interact, vol. 3, no. 1, pp. 91–122, 2014.

[29]. Kupferberg A, Glasauer S, Huber M, Rickert M, Knoll A, and Brandt T, “Biological movement increases acceptance of humanoid robots as human partners in motor interaction,” AI Soc, vol. 26, no. 4, pp. 339–345, 2011.

[30]. Destephe M, Maruyama T, Zecca M, Hashimoto K, and Takanishi A, “Improving the human-robot interaction through emotive movements A special case: Walking,” ACM/IEEE Int. Conf. Human-Robot Interact, 2013.

[31]. Bisio A, Sciutti A, Nori F, Metta G, Fadiga L, Sandini G, and Pozzo T, “Motor contagion during human-human and human-robot interaction,” PLoS One, vol. 9, no. 8, p. e106172, 2014. [PubMed: 25153990]

[32]. Begum M, Huq R, Wang RH, and Mihailidis A, “Collaboration of an assistive robot and older adults with dementia,” Gerontechnology, vol. 13, no. 4, pp. 405–419, 2015.

[33]. Bovbel P and Nejat G, “Casper: An assistive kitchen robot to promote aging in place,” J. Med. Device, vol. 8, no. 3, 2014.

[34]. Das B, Cook DJ, Krishnan NC, and Schmitter-Edgecombe M, “One-Class Classification-Based Real-Time Activity Error Detection in Smart Homes,” IEEE J. Sel. Top. Signal Process, vol. 10, no. 5, 2016.

[35]. Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, and Ng AY, “ROS: An open-source Robot Operating System,” ICRA Work. Open Source Softw, vol. 3, no. 3.2, p. 5, 2009.

[36]. Eaton E, Mucchiani C, Mohan M, Isele D, Luna JM, and Clingerman C, “Design of a low-cost platform for autonomous mobile service robots,” in IJCAI Workshop on Autonomous Mobile Service Robots.

[37]. Boren J and Cousins S, “The SMACH high-level executive,” IEEE Robot. Autom. Mag, vol. 17, no. 4, pp. 18–20, 2010.

[38]. Hess W, Kohler D, Rapp H, and Andor D, “Real-time loop closure in 2D LIDAR SLAM,” in IEEE International Conference on Robotics and Automation, 2016, pp. 1271–1278.

[39]. Thrun S, Burgard W, and Fox D, Probabilistic Robotics. MIT Press, 2005.

[40]. Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, and Murphy K, “Speed/accuracy trade-offs for modern convolutional object detectors,” CoRR, vol. abs/1611.1, 2016.

[41]. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, and Berg AC, “SSD: Single shot multibox detector,” in European Conference on Computer Vision, 2016, pp. 21–37.

[42]. Broadbent E, Stafford E, and MacDonald B, “Acceptance of healthcare robots for the older population: Review and future directions,” Int. J. Soc. Robot, vol. 1, no. 4, p. 319, 2009.

[43]. Wilson C, Hargreaves T, and Hauxwell-Baldwin R, “Smart homes and their users: A systematic analysis and key challenges,” Pers. Ubiquitous Comput, vol. 19, no. 2, pp. 463–476, 2015.

[44]. Chen L, Cook DJ, Guo B, Chen L, and Leister W, “Guest editorial: Special issue on situation, activity, and goal awareness in cyber-physical human-machine systems,” IEEE Trans. Human-Machine Syst, vol. 47, no. 3, 2017.

[45]. Cook DJ, Crandall AS, Thomas BL, and Krishnan NC, “CASAS: A smart home in a box,” Computer (Long. Beach. Calif)., vol. 46, no. 7, 2013.

[46]. Rafferty J, Nugent CD, Liu J, and Chen L, “From activity recognition to intention recognition for assisted living within smart homes,” IEEE Trans. Human-Machine Syst, vol. 47, no. 3, pp. 368–379, Jun. 2017.

[47]. Gayathri KS, Easwarakumar KS, and Elias S, “Probabilistic ontology based activity recognition in smart homes using Markov Logic Network,” Knowledge-Based Syst, vol. 121, pp. 173–184, Apr. 2017.

[48]. Cook DJ, "Learning setting-generalized activity models for smart spaces," IEEEIntell. Syst, vol. 27, no. 1, 2012.

[49]. Brenon A, Portet F, and Vacher M, "Using statistico-relational model for activity recognition in smart home," in European Conference on Ambient Intelligence, 2015.

[50]. Fang H, Si H, and Chen L, "Recurrent neural network for human activity recognition in smart home," in Chinese Intelligent Automation Conference, 2013, pp. 341–348.

[51]. Reiss A, Stricker D, and Hendeby G, "Towards robust activity recognition for everyday life: Methods and evaluation," in Pervasive Computing Technologies for Healthcare, 2013, pp. 25–32.

[52]. Das B, Cook DJ, Krishnan N, and Schmitter-Edgecombe M, "One-class classification-based real-time activity error detection in smart homes," IEEE J. Sel. Top. Signal Process, 2016.

[53]. Estimote, "The physical world. Software-defined," 2018 [Online]. Available: https://estimote.com/.

[54]. COCO, "Overview," Common objects in context, 2018 [Online]. Available: http://cocodataset.org/#overview.

[55]. Peng C, Xiao T, Li Z, Jiang Y, Zhang X, Jia K, Yu G, and Sun J, "MegDet: A large mini-batch object detector," in IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6181–6189.

[56]. Lewis JR, "Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ," in Human Factors Society 36th Annual Meeting, 1992, pp. 1259–1263.

[57]. Chin K, Diehl J, Norman V, "Development of an instrument measuring user satisfaction of the human-computer interface," in International Conference on Human Factors in Computing Systems, 1988, pp. 213–218.

[58]. Hone KS and Graham R, "Subjective assessment of speech-system interface usability," in Eurospeech, 2001.

[59]. Ghallab M, Nau D, and Traverso P, Automated Planning and Acting. Cambridge, UK: Cambridge University Press, 2016.

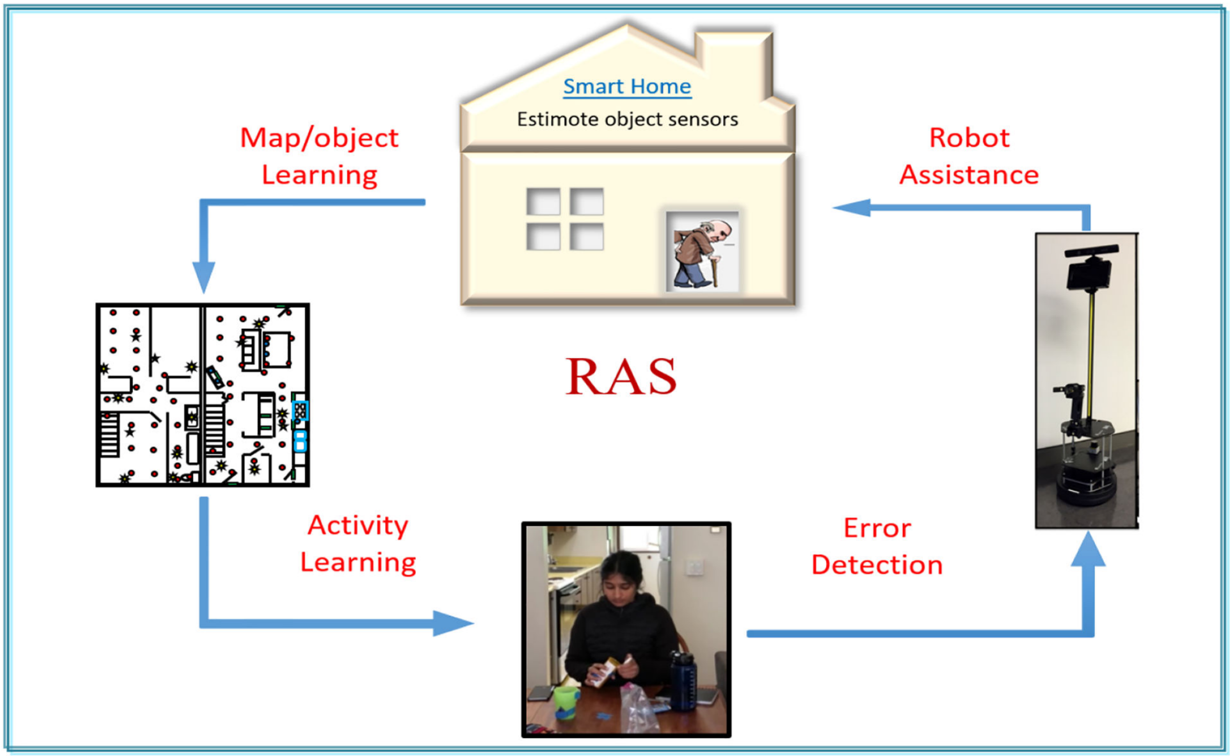


Figure 1.
RAS system overview.

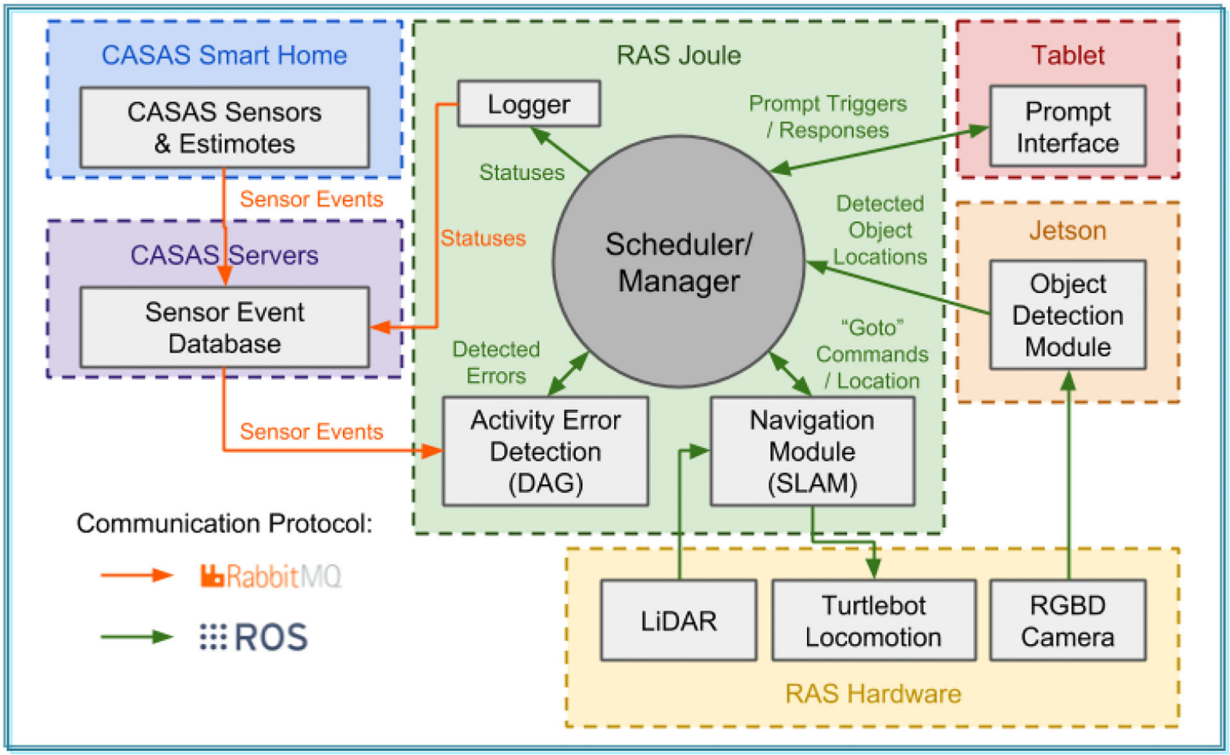


Figure 2.

RAS software architecture. Components are located physically within a CASAS smart home (blue) or on a CASAS server (purple), RAS Joule (green), RAS hardware (yellow), Jetson (orange), or a tablet interface (red). Components communicate via a RabbitMQ (orange arrows) or through ROS (green arrows).

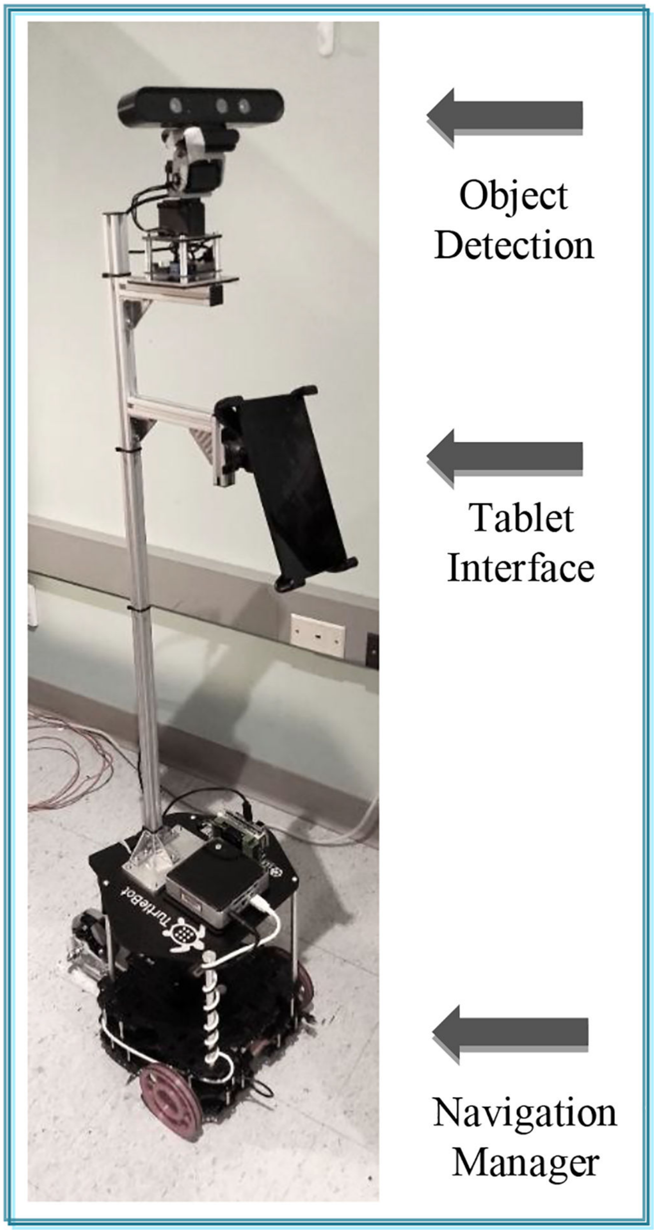


Figure 3.

The Turtlebot 3 is enhanced with sensors, camera, computer, navigation system, and Android tablet for use in RAS.

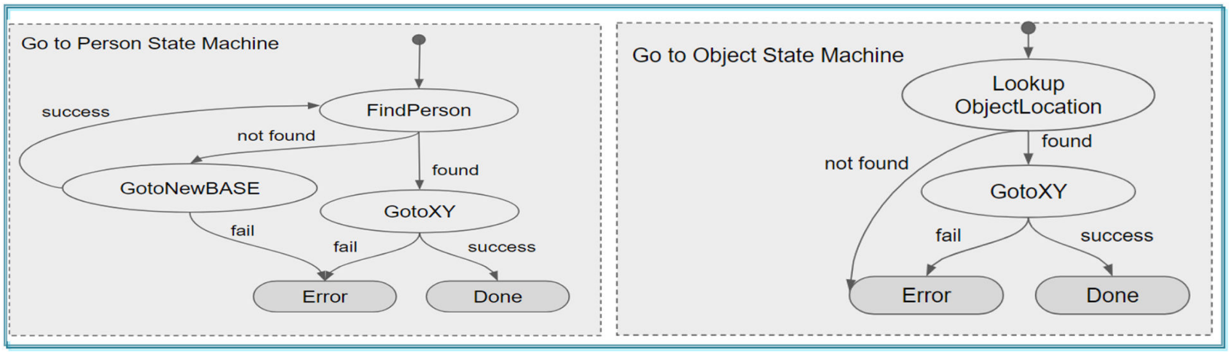


Figure 4.

The two RAS state machines. (left) The Go to Person SMACH. (right) The Go to Object SMACH.

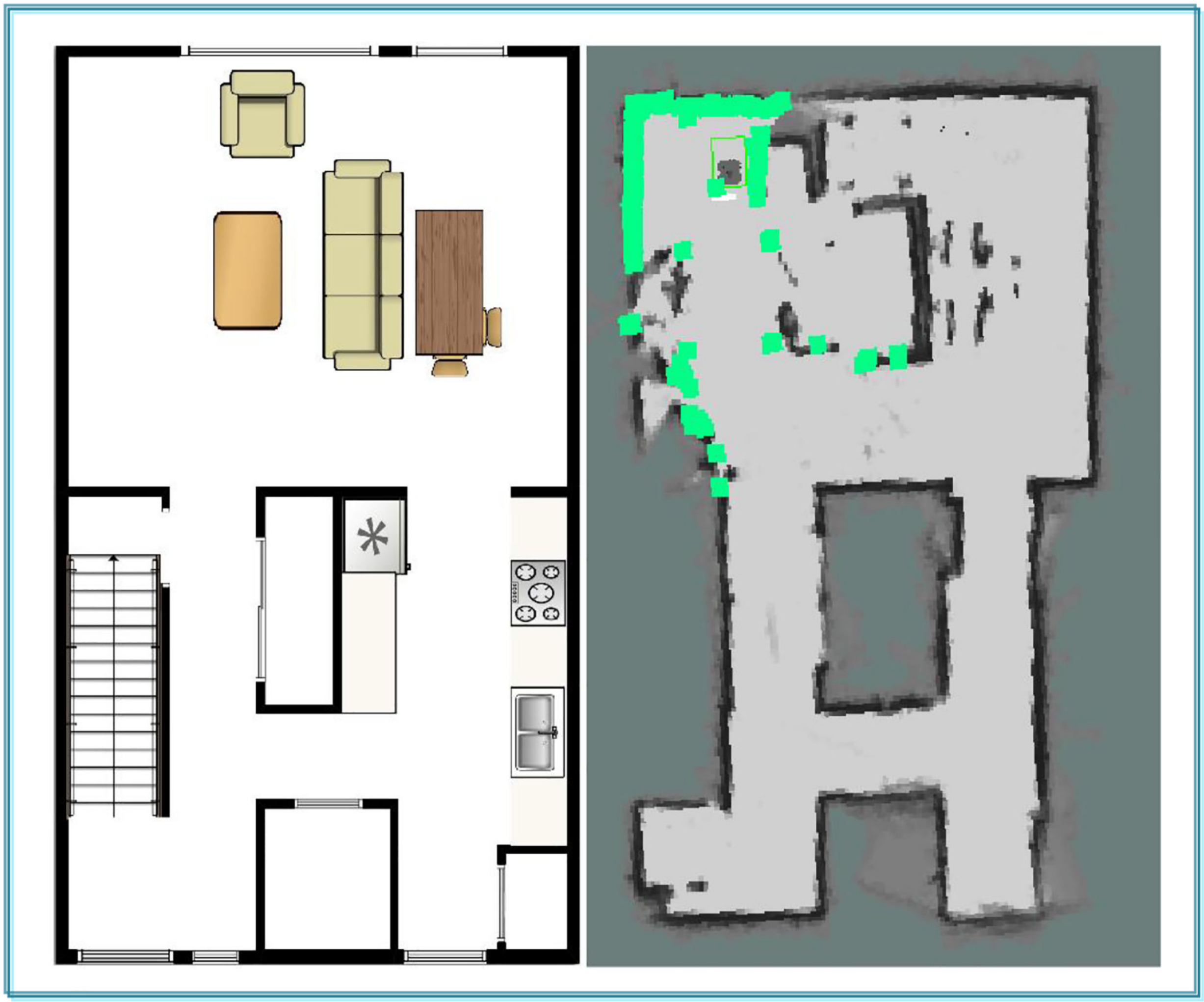


Figure 5.

(left) Smart home floorplan. (right) Cartographer-generated home map. Green dots indicate current LiDAR scan.

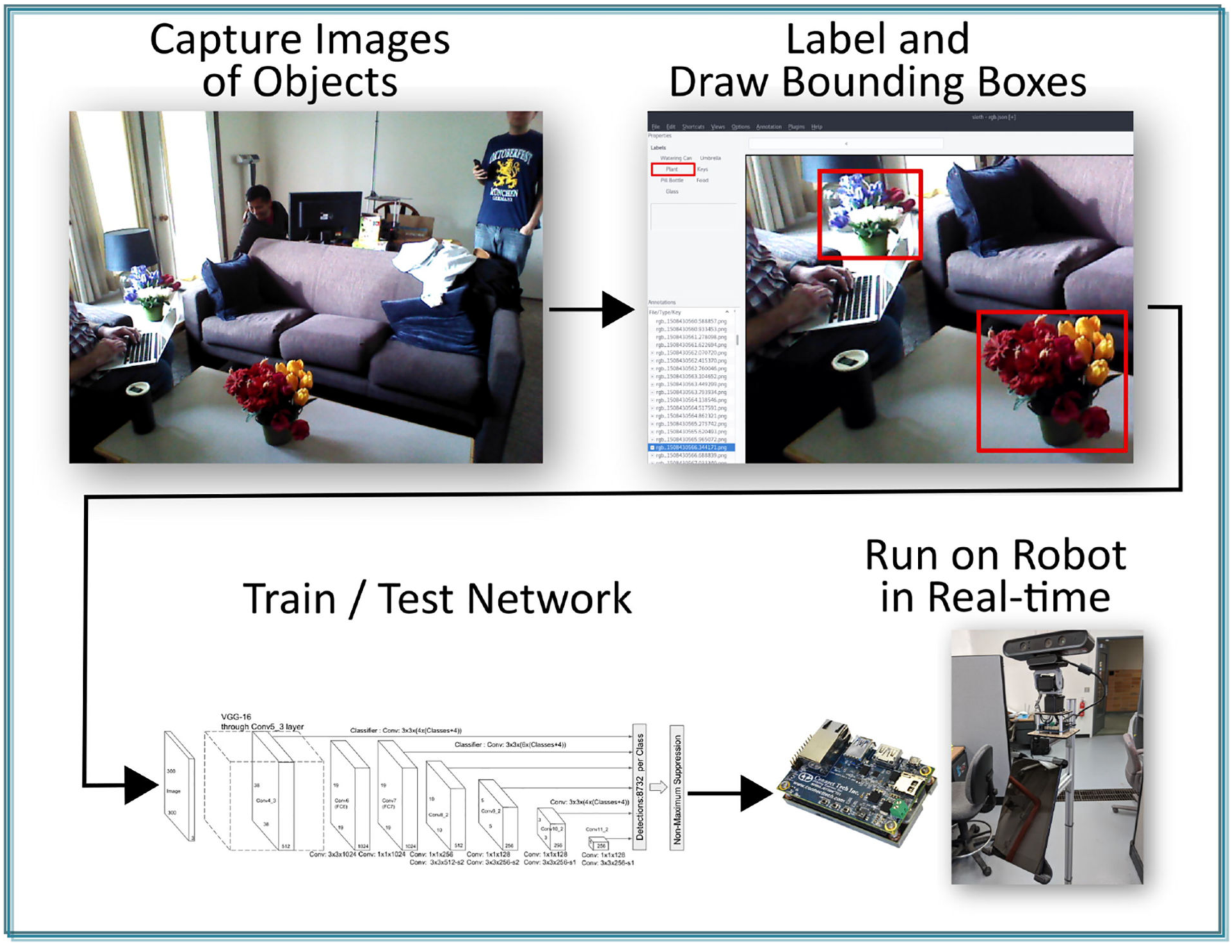


Figure 6.

Object detection process. (upper left) Images containing objects of interest are captured from the home or image repositories. (upper right) Training images are preprocessed by labeling objects and drawing corresponding bounding boxes. (lower left) The labeled images are used to train a CNN. (lower right) The trained network is exported to RAS to use for real-time object detection in smart home environments.

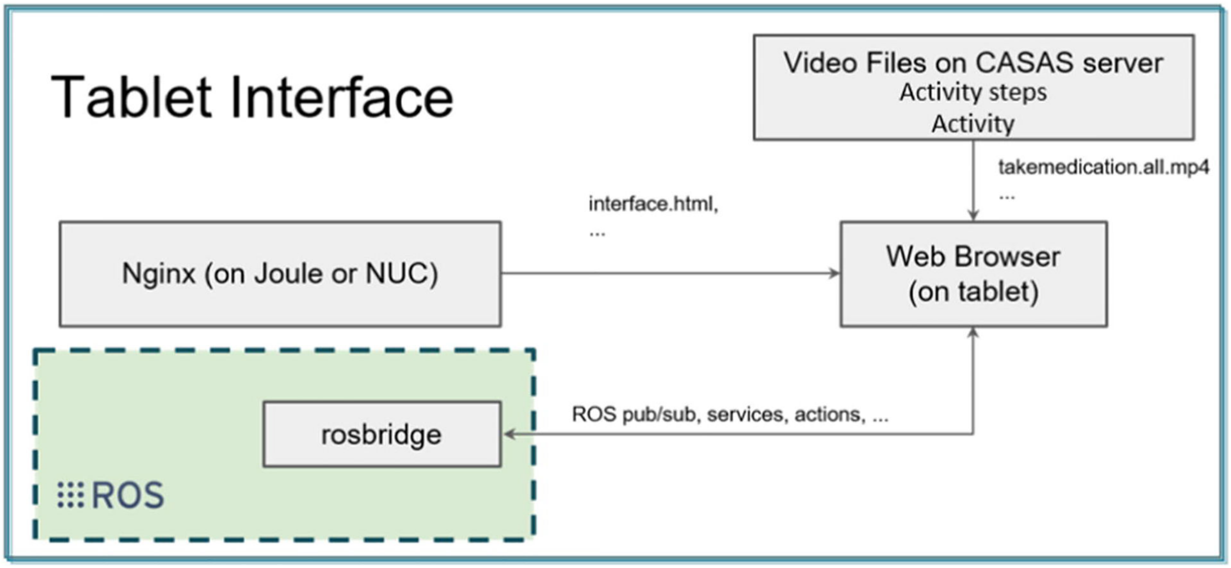
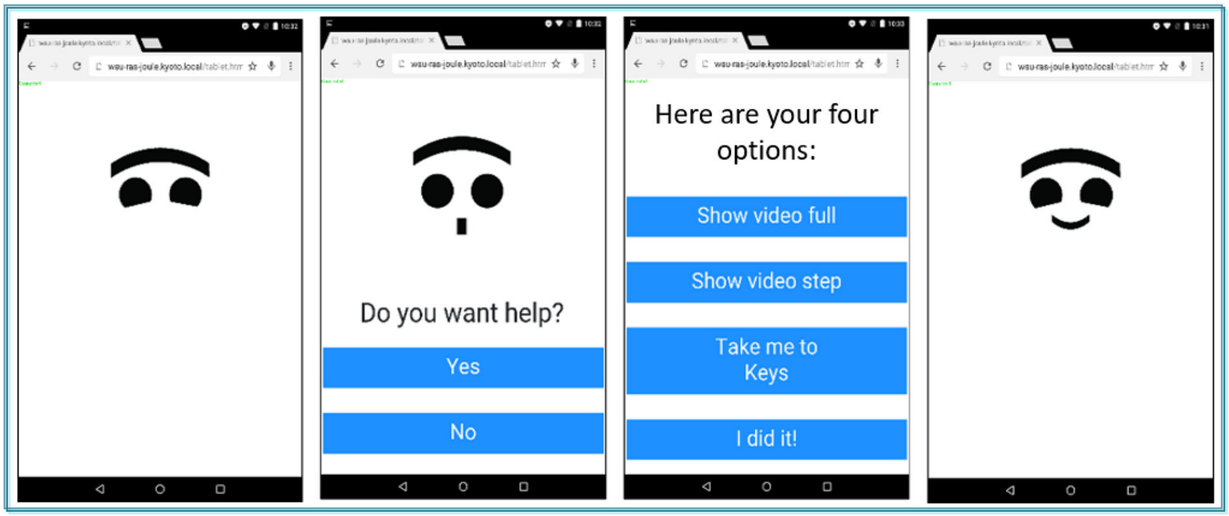


Figure 7.
RAS user interface module connections between tablet, database, and web site.



Do you want help?

Here are your four options:

Figure 8.

RAS user interface states. (from left to right) Neutral state expression is displayed while observing activity; questioning expression; assistance options; pleased expression.

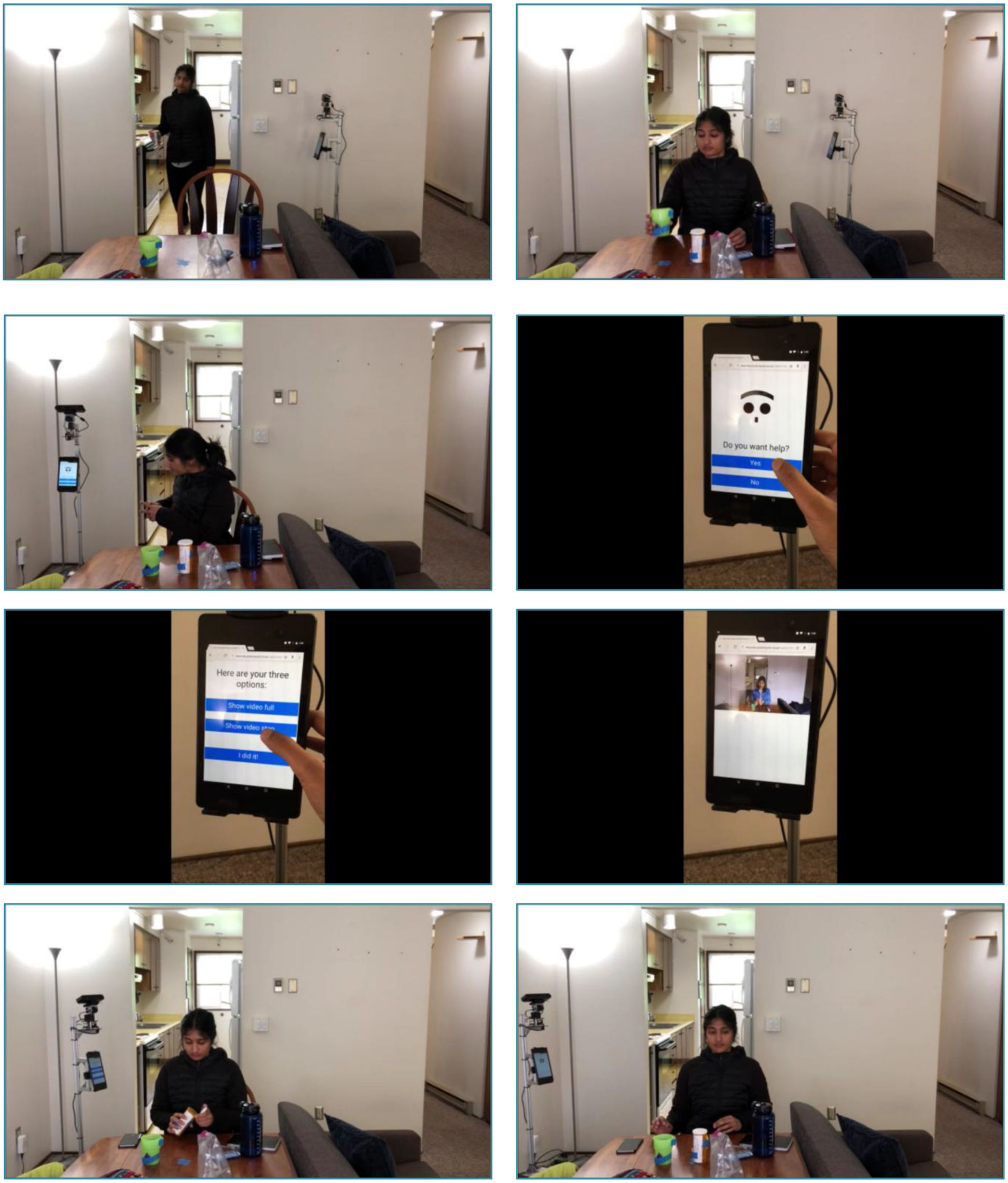


Figure 9.

RAS helps s smart home resident complete the “take medicine with food and water” activity. (first row left) RAS robot observes activity, (first row right) resident skips “take medicine” step, (second row left) robot approaches resident, (second row right) robot offers help, (third row left) resident asks robot to show current activity step, (third row right) video shows person taking medicine, (fourth row left) resident takes medicine and completes missing step, (fourth row right) robot interface looks pleased after step is completed.

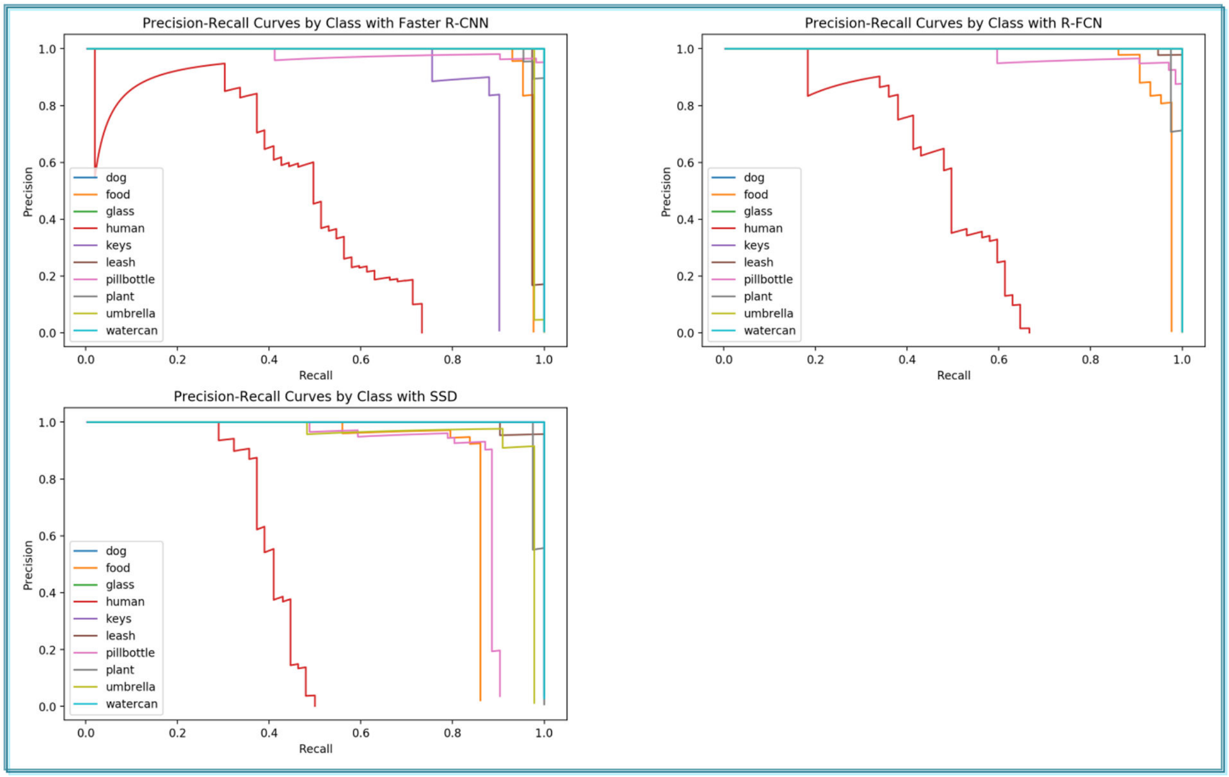


Figure 10.

Precision-recall curves by class for the R-CNN, R-FCN, and SSD networks.

Table 1.

List of steps for each activity. Participants completed each activity four times, once completely and once for each type of error.

Activity: Walk the dog Tagged objects: umbrella, leash, keys, dog, front door	
Retrieve umbrella from closet	Error 1: skip this step
Retrieve leash from hall hook	Error 2: skip this step
Retrieve keys from kitchen	Error 3: skip this step
Put leash on dog	
Leave home with dog and items	
Activity: Take medication with food Tagged objects: granola bar, cup, pill bottle, dining chair, sink, garbage door	
Retrieve granola bar from kitchen	Error 1: skip this step
Retrieve cup from kitchen	
Fill cup with water from sink	
Take water and granola bar to dining table	
Retrieve medication from kitchen	Error 2: skip this step
Take medication to dining table	
Eat granola bar and put it down	
Take two medicine tablets	Error 3: skip this step
Drink water	
Put medication bottle and water cup back, throw away wrapper	
Activity: Water plants Tagged objects: watering can, sink, coffee table plant, side table plant	
Retrieve water can from living room	
Fill can with water from kitchen sink	Error 1: skip this step
Water plant on coffee table	Error 2: skip this step
Water plant on side table	Error 3: skip this step
Empty can of water and put back	

Table 2.

Class sizes and average precision for each object class for each of the object detection networks.

Object class										
# examples	**Leash**	**Dog**	**Keys**	**Cup**	**Pill bottle**	**Food**	**Umbrella**	**Plant**	**Watering can**	**Human**
manually collected	230	457	195	556	1,509	849	389	661	725	1,347
after sampling	70,588	72,484	68,967	75,861	93,485	81,530	73,841	78,101	79,216	94,449

Object class										
Network	**Leash**	**Dog**	**Keys**	**Cup**	**Pill bottle**	**Food**	**Umbrella**	**Plant**	**Watering can**	**Human**
R-CNN	0.99	1.00	0.90	1.00	0.96	0.69	0.98	0.99	1.00	0.47
R-FCN	0.97	1.00	1.00	1.00	0.97	0.97	1.00	0.96	1.00	0.46
SSD	1.00	1.00	1.00	1.00	0.87	0.94	0.96	0.99	1.00	0.43

Table 3.

Participant questionnaire data (mean and standard deviation values).

Questionnaires			
PSSUQ			
	Overall	4.55(1.95)	1.21–6.95
	System Usefulness	4.37(1.89)	1.13–6.88
	Interface Quality	4.83 (1.94)	1.67–7.00
	Information Quality	4.66 (2.23)	1.00–7.00
QUIS			
	Screen	7.96 (0.97)	6.00–9.00
	Terminology	7.74(1.42)	3.67–9.00

Notes: PSSUQ = Post-Study System Usability Questionnaire; QUIS = Questionnaire for User-Interface Satisfaction.

Table 4.

Participant ratings (mean and standard deviation values) of each prompt type for four separate purposes.

	Prompt Type		
Prompt Purpose	**Guide to Object**	**Next Step Video**	**Full Video**[**]
Ease of Imitation	3.88 (0.95)	4.08 (0.88)[a]	3.50 (1.14)
Range	(2–5)	(1–5)	(1–5)
Helpful if Unable to Recall Next Step	4.08 (0.78)	4.21 (0.66)[a]	3.55 (1.22)
Range	(2–5)	(2–5)	(1–5)
Confusing to Someone with MCI[*]	2.67 (1.09)	2.25 (0.90)[ab]	2.86 (1.08)
Range	(1–5)	(1–5)	(1–5)
Helpful to Someone with MCI	4.04 (0.81)[a]	4.08 (0.58)[a]	3.41 (1.14)
Range	(2–5)	(3–5)	(1–5)

Notes: N = 24, the number of participants who answered these questions, unless otherwise indicated; MCI = mild cognitive impairment

[*] Lower scores should be interpreted as less confusing;

[**] Represents analyses completed with N= 22 participants who answered these questions;

[a] Represents significant difference from full video;

[b] Represents significant difference from guide to object.

Table 5.

Proportion of participants' ratings for most liked, most helpful and least effective prompt styles.

	Prompt Type		
Rating Category	**Guide to Object**	**Next Step Video**	**Full Video**
Most Liked Prompt	32%	60%	8%
Most Helpful Prompt	40%	52%	8%
Least Effective Prompt	12%	8%	80%

Notes: N = 25; one participant did not answer. Chi-square tests of independence revealed significant differences among prompt preferences for each rating category, χ^2s (2, N = 25) < 7.00, ps < .022.