applied sciences

MDPI

Article

Functionalities, Benchmarking System and Performance Evaluation for a Domestic Service Robot: People Perception, People Following, and Pick and Placing

Meysam Basiri [1,*], João Pereira [1], Rui Bettencourt [1], Enrico Piazza [2], Emanuel Fernandes [1], Carlos Azevedo [1] and Pedro Lima [1]

[1] Institute for Systems and Robotics, Tecnico Lisboa, 1049-001 Lisboa, Portugal; 95joaopereira@gmail.com (J.P.); rui.bettencourt@tecnico.ulisboa.pt (R.B.); emanuel.a.fernandes@tecnico.ulisboa.pt (E.F.); cguerraazevedo@tecnico.ulisboa.pt (C.A.); pedro.lima@tecnico.ulisboa.pt (P.L.)

[2] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy; enrico.piazza@polimi.it

* Correspondence: meysam.basiri@tecnico.ulisboa.pt

Abstract: This paper describes the development of three main functionalities for a domestic mobile service robot and an automatic benchmarking system used for the systematic performance evaluation of the robot's functionalities. Three main robot functionalities are addressed: (1) People Perception, (2) People Following and (3) Pick and Placing, where the hardware and software systems developed for each functionality are described and demonstrated on an actual mobile service robot, with the goal of providing assistance to an elderly person inside the house. Furthermore, a set of innovative benchmarks and an automatic performance evaluation system are proposed and used to evaluate the performance of the developed functionalities. These benchmarks are now made publicly available and is part of the European Robotics League (ERL)-Consumer to systematically evaluate the performance of service robot solutions at different testbeds around Europe.

Keywords: domestic service robots; benchmarking; performance evaluation system; pick and placing; human perception; human following

Citation: Basiri, M.; Pereira, J.; Bettencourt, R.; Piazza, E.; Fernandes, E.; Azevedo, C.; Lima, P. Functionalities, Benchmarking System and Performance Evaluation for a Domestic Service Robot: People Perception, People Following, and Pick and Placing. *Appl. Sci.* **2022**, *12*, 4819. https://doi.org/10.3390/app12104819

Academic Editors: Mihai Andries, Plinio Moreno and Alexandre Bernardino

Received: 14 March 2022
Accepted: 4 May 2022
Published: 10 May 2022

1. Introduction

Mobile robot systems are now being successfully deployed in different application domains such as manufacturing, medicine, logistics and search and rescues [1–4]. However, in terms of domestic service applications, mostly simple systems such as floor and pool cleaning have reached maturity [5]. Domestic assistance, in particular for handicapped and elderly people, is still a great challenge of our society where mobile robots can be significantly beneficial. Domestic environments are complex, namely due to their dynamic and unstructured conditions and the need for a close interaction with objects [6] and humans [7]. This complexity arises from a wide variety of challenges: perception [8], navigation [9], HRI [10], manipulation [11] and task planning [12]. A general purpose service robot for a domestic environment is required to handle different tasks such as setting the table, serving food, cleaning the house, interacting with humans, finding and picking up objects, recognizing and handling visitors. A survey of robot systems integrating some of those capabilities in RoboCup@Home competitions can be found in [13]. To perform such complex tasks, the robot needs to hold and employ a variety of smaller specific skills (referred to here as functionalities), such as navigation, speech recognition, object perception and manipulation, where a set of hardware/software modules are utilized for each functionality.

Quantitative evaluation of a robot's ability to perform a specific task or functionality, in real-world conditions, is currently considered an open challenge. Different research

efforts [14] have recommended protocols and methods for an evidence-based evaluation of robotics research. When interaction with the environment is needed, such as interaction with people or objects, a solution is still missing that can guarantee experiment *reproducibility*, i.e., the possibility of repeating the experiment by other independent researchers to verify the results, and *repeatability*, i.e., the characteristic of an experiment that gives the same outcome when executed at different locations or different times.

This paper describes the implementation of three key functionalities for a domestic service robot, *People perception*, *People following* and *Pick and placing*, while proposing three innovative functionality benchmarks and an autonomous performance evaluation system for systematic evaluation of these three functionalities. These benchmarks are now integrated into the benchmarking system of the European Robotics League (ERL-Consumer) [15], which aims at systematic performance evaluation of robot solutions in the format of competitions [16]. The ERL-Consumer challenge occurs frequently at different testbeds around Europe and covers the domain of consumer service robotics, with current focus on applications for the benefit of addressing societal challenges such as healthy aging and longer independent living.

The structure of this paper is as follows: Section 2 introduces some of the related work on robot benchmarking and the targeted functionalities. Section 3 describes the implementation of three robot functionalities for an actual domestic service robot , while describing in detail the hardware and software systems used for each functionality. Section 4 presents three functionality benchmarks for the systematic performance evaluation of the three robot functionalities and presents an automatic benchmark control and performance evaluation system for the quantitative evaluation of robot functionalities. Section 5 presents experiments and results from trials where the benchmarks and the benchmarking system are used to evaluate the functionalities presented in Section 3. Finally, Section 6 concludes the paper with the conclusion and future works.

2. Related Work

Over the recent years, advanced service robot systems are rapidly growing in domestic applications, capable of performing specific skills or functionalities, such as manipulation [17] or object recognition [18], or complex global tasks [13]. The emergence of many different solutions has led to the need for a more systematic evaluation of the performance of robotic systems [19] and is triggering initiatives to create benchmarks and standard experimental testbeds. Our approach to benchmarking robot solutions is based on the definition of two types of benchmarks that were originally introduced in a former project, RoCKIn [19,20]:

- Functionality Benchmarks (FBMs): evaluates the performance of hardware and software systems dedicated to a single and specific functionality, in the context of experiments focused on that functionality. Examples: Navigation FBM and Object Perception FBM;
- Task Benchmarks (TBMs): evaluate the performance of integrated robot systems executing complex tasks that need the interaction/composition of different functionalities. Examples: Welcoming Visitors TBM and Cleaning the house TBM.

In our previous work, we have demonstrated a set of TBMs and FBMs in the context of Service Robot competitions and introduced FBMs such as Navigation, Object Perception and Speech Understanding [16]. In this paper, we will introduce three additional FBMs while developing and evaluating three key functionalities for a domestic service robot.

Human–robot interaction (HRI) is one of the most important capabilities that a domestic service robot needs to hold, of which People perception is the first key element for having a successful HRI. People perception can be divided into several sub-problems: (1) detecting the presence of a person, (2) estimating the person's position and (3) recognizing the identity of the detected person. To detect a person, many different sensors and approaches are available today, ranging from simple laser-based leg detectors [21,22] to detecting heat signatures through a thermal camera [23] or other computer vision tech-

niques [24]. Upon detection, the person's position in the world can be estimated using the depth component of an RGB-D camera, laser sensors [25] or using a stereo camera [26] or multi-view images. Finally, for person recognition or people re-identification, the state-of-the-art methods mostly rely on vision sensors and deep learning methods trained from RGB images or videos [27,28] but also using other sensors such as depth cameras [29] or infrared cameras [30] to overcome visual limitations such as varying light conditions or changes in clothing, amongst others.

In the context of HRI, in particular for domestic and consumer applications, the ability for a robot to effectively follow a human is significantly important. For example, to follow a child, the elderly or a visitor inside the house or to carry the bags of a person inside an airport or a retail store. The people-following problem is commonly addressed by (1) detecting and tracking a target person in the environment and (2) controlling the movement of the robot to maintain a desired distance with the target person. Most solutions for people following addressed in the literature rely on visual cues [31,32], however, attempts using other type of sensors such as laser scanners or a combination of sensors are also available [33].

Pick and place manipulation is yet another important capability that is crucial for service and domestic service robots [34], for example, to set up a dining table or find and deliver objects in the house. Such capability is tightly linked to the object detection and recognition functionality and would require a robust image processing algorithm [35]. A pick and place routine usually consists of four main steps [36]: (1) choosing a grasp on the item, (2) planning a motion to grasp the item, (3) planning a motion to the placement location and (4) plan a motion to extract the gripper. The visual servoing method [37] has been widely used in the literature to integrate visual information in the robot arm's control loop. Its fundamental idea is to continuously estimate the pose of the target object relative to the end-effector using a camera. This pose is used as an error value to be minimized by a control law, which prescribes the manipulator motion necessary to approach the target object. Several variants of this technique have been developed, including the recent direct visual servoing approach which considers the complete image as the input (instead of artificial markers) by calculating its luminance map [38] or by feeding it to a deep neural network [39]. Neural networks have also been applied to the grasp-selection problem to achieve firm grasps on unknown objects [40]. Fully end-to-end reinforcement learning systems have been developed [41,42] which are able to learn complete grasping policies based only on RGB images and using a substantial amount of experience data gathered from multiple robots running simultaneously.

3. Robot Functionalities: People Perception, Human Following and Pick and Placing

This section describes the hardware and software systems used for developing three main robot functionalities for a domestic mobile service robot. Section 3.1 briefly describes the mobile robot platform developed and upgraded for the intended research work as well as for participating in major robot competitions such as ERL and RoboCup@Home, while the later subsections will present the three functionalities in detail:

- People perception (Section 3.2);
- Human following (Section 3.3);
- Pick and placing (Section 3.4).

3.1. Mobile Robot System

The MOnarCH robot (mbot), shown in Figure 1, originally designed to interact with children inside hospitals [9] and to participate at domestic robot competitions [43], was improved and used to develop the three functionalities presented in this paper. In addition to various other sensors and actuators, described in [9], the robot is now equipped with 25 m range laser scanners, which are used for mapping, navigation and obstacle avoidance and a display with touch screen. On top of this platform, we have now installed additional devices, namely, a 6 DoF arm for manipulation (Kinova Gen2), a directional microphone for

speech interaction (Røde VideoMic Pro) and an Orbbec Astra RGB-D camera positioned on the (rotating) head for object detection and localization, people tracking, obstacle perception and visual servoing. The mobile robot features one on-board computer with i7 processor, graphics card NVIDIA GTX 1060 AERO. It has an overall weight of 25 Kg and a maximum velocity of 2.5 m/s with 1 m/s^2 acceleration. The system includes WiFi communications, which, among other purposes, enable it to communicate with a home automation system to send remote commands and read remote sensors. All the code running in the mbot is written in Python and C++, supported by the Robot Operating System (ROS) and some of its packages.

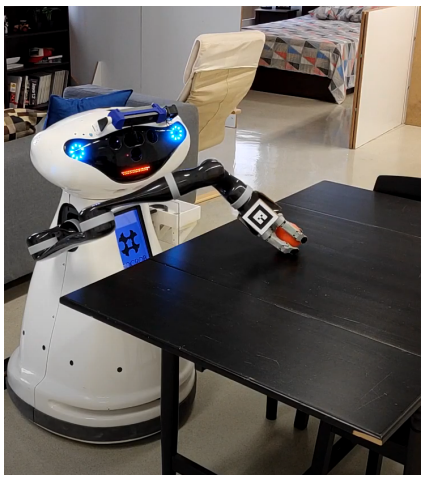


Figure 1. The Mbot mobile robot at the ISR/IST ERL certified test bed performing a pick and place operation.

3.2. People Perception

People perception is an important functionality required for robots operating in proximity of humans, in particular for robots in home and consumer environments expected to interact with humans. To obtain an efficient and complete people perception functionality, we apply and merge several techniques that allow the robot to (1) detect the presence of a person, (2) to locate the position of the person and to (3) identify the identity of the detected person.

In this work, we employed a technique common to both detecting people and objects based on the Darknet YOLO [44] method. This method allows the detection in real-time of multiple people by returning a 2D bounding box of the detected people in the camera frame. The method receives RGB images from the RGB-D camera and uses the full images as input to a pre-trained neural network, resulting in an image that is divided into several regions and predicting possible bounding boxes with associated probabilities for each region.

Upon detection, an estimate of the person's 3D position is computed. This is achieved by using the information registered by the RGB-D camera that supplies both RGB and depth images. These two images are synchronized (≈12 Hz), then the corners of the detected bounding box are found in the depth image. The bounding box is shrunk to increase the density of interest points, and the center of this bounding box is defined as an approximation to the detected person's position [45].

For people recognition, the used method is based on face recognition techniques [46]. The algorithm receives sampled frames from the RGB-D camera and feeds the image to a pre-trained neural network that detects all faces in an image by using the method Histogram of Oriented Gradients [47]. When all faces in an image are detected, each face has 68 face landmarks estimated [48] so that the face can be rotated and scaled to be centered in a square. This image is then fed to one other pre-trained neural network that returns an embedding of 128 measurements of each face [49] that can be used to compare and recognize faces. Then, for each image, the obtained embedding is compared with the

previously known faces by applying a linear Support-Vector Machine (SVM) Classifier that returns the name of the person with closest embedding match, above a certain tolerance threshold. Algorithm 1 describes the implementation of this method while Figure 2 shows an instance from one of the face recognition trials.

Algorithm 1: Face detection algorithm.

```
Result: Face detected from a group of known faces
input: rgb_image, known_faces_list[]
1  begin
2      frame = Sample(rgb_image);
3      face_list = face_detection_nn(frame);
4      for face in face_list do
5          face_features = features_estimator(face);
6          face_centered = center(face, face_features);
7          embedding = face_encodings(face_centered);
8          face_estimate = classifier(embedding, known_faces_list[]);
9      end
10 end
```

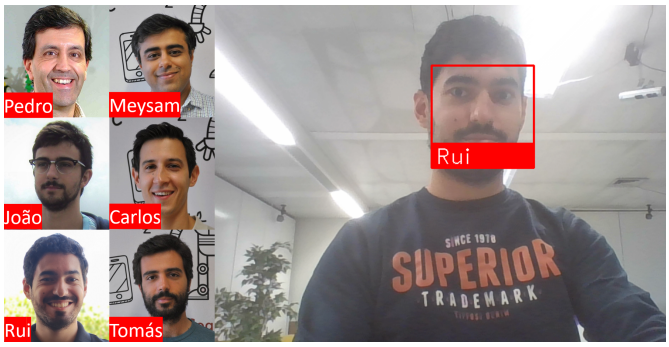


Figure 2. Face recognition of an encountered person (on the **right**) from a pool of 6 known faces (on the **left**).

3.3. Person Following

The Person Following module requires that the People Perception component, described in Section 3.2, is working correctly and a person can be correctly detected, identified and his/her location can be accurately estimated. Our people following method is built on top of an autonomous waypoint navigation method previously developed for our robot [45]. The base navigation is the ROS navigation stack (http://wiki.ros.org/navigation, (accessed on 1 September 2020)) with AMCL (http://wiki.ros.org/amcl, (accessed on 15 September 2020)), a particle filter based localization implementation. For motion planning, a Dijkstra based global planner is used as well as the Dynamic Window Approach used as the local planner.

Algorithm 2 describes the Person Following routine. After obtaining the position of the target person relative to the robot's base and, consequently, relative to the map's frame, the algorithm calculates the desired goal pose where the robot should be located at. This goal pose is then fed to the autonomous navigation stack that guides the robot to the target position taking the shortest path while also avoiding obstacles. If the target person moves positions, then the robot will track the movement and reapply the algorithm.

Steps for calculating the goal pose (lines 5 to 17 of the Algorithm 2) can be summarized in the following manner, where r is the desired distance that the robot should keep from the person:

1. A circle of radius r, number of points n_{pc} and density of points d_p is created around the target ($n_{pc} = d_p r$).

2. All the points are ordered by the distance to the robot.
3. The points are tested to check if they are free in the costmap.
4. Check if there is any obstacle between the robot and the target.
5. If there are points in this circle that satisfy these conditions, the closest point to the robot is used. If there is not, r increases and the algorithm returns to step 1.
6. The orientation of the robot is chosen from the calculated position towards the person.

Algorithm 2: Person following algorithm

```
Result: The robot follows a certain person.
input: target_person_pose, robot_pose, map, r, d_p
begin
    while people_following is true do
        if person_moved(target_person_pose, old_person_pose) then
            target_pose = ø
            while target_pose = ø do
                circle[] = circle_creation(target_person_pose, r, d_p);
                ordered_circle[] = order(circle[], robot_pose);
                ordered_circle[] = check_availability(ordered_circle[], map);
                if ordered_circle is not empty then
                    target_pose = ordered_circle[first_element];
                    orientation = get_orientation(target_pose, target_person_pose);
                    break;
                end
                else
                    r = r + increase_constant;
                end
            end
            Move_To(target_pose, orientation);
            old_person_pose = target_person_pose;
        end
        rotate_head_towards(target_person_pose);
    end
end
```

Figure 3 illustrates an example of people following from an actual trial where the robot selects the best possible goal position while navigating through a narrow corridor.

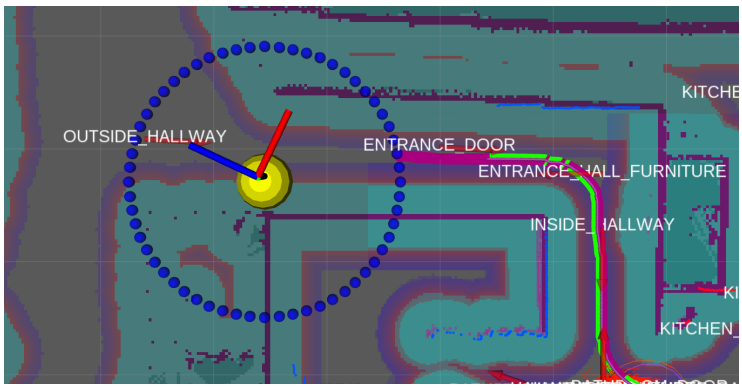


Figure 3. Person following example. The yellow sphere represents the person position, blue spheres represent the possible positions the robot could be at the desired distance to the person and the purple line is the path the robot chose from its position to the chosen target position.

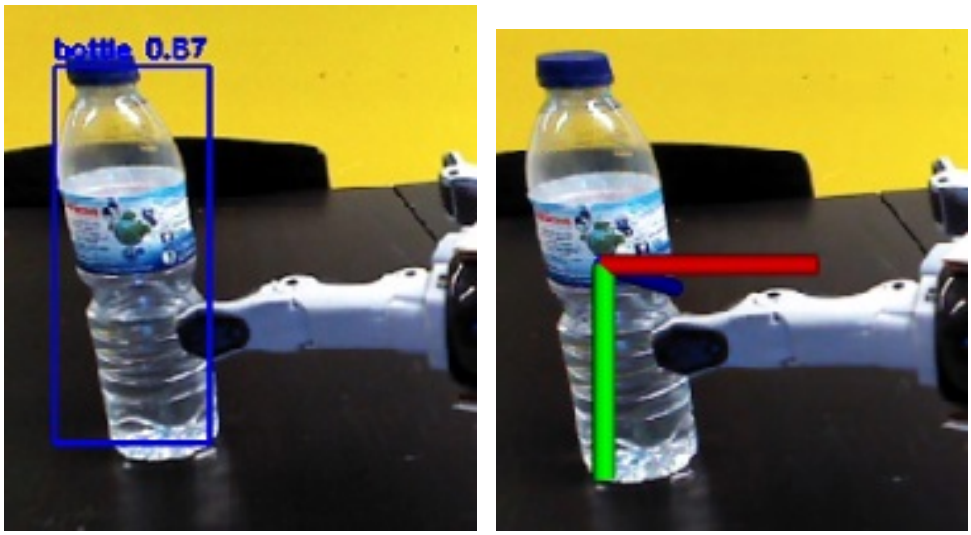
3.4. Pick and Placing

For the Pick and Placing functionality, an approach based on the visual-servoing technique is implemented for reaching and grasping objects. The method relies on the RGB-D camera in the MBot's head to continuously estimate both the end-effector and target poses and then uses their difference as a positioning error to be minimized through proportional control of the arm. Since both poses are in the camera frame, this approach is less sensitive to errors in the calibration of both the camera and the arm joints.

3.4.1. Target Object Localization

Although traditional visual-servoing implementations use artificial markers on the target object, recent advances in image recognition allow for regular objects to be detected without markers. We use a YOLO v3 [44] Convolutional Neural Network, trained on the COCO dataset containing images of 80 object classes. The network outputs a 2D bounding box around detected objects in the camera frame (as shown in Figure 4a) at a high refresh rate (>10 Hz).

To estimate the 3D pose of an object, the central portion of the bounding box is sampled, and its depth values are obtained. The 25th percentile depth value is selected as representative of the object depth d. This percentile was chosen to remove potential outliers. A ray is then projected from the camera lens and intersects the center of the YOLO bounding box. This ray is obtained through a pinhole camera model and is represented by unit vector $\mathbf{r}$. The 3D object pose $\hat{g}^c$ (Figure 4b) is obtained by multiplying this direction vector by the depth value: $\hat{g}^c = d \cdot \mathbf{r}$.



(**a**) (**b**)

Figure 4. Target localization: (**a**) YOLO bounding box; (**b**) 3D pose obtained through depth sampling.

3.4.2. End-Effector Localization

To obtain the end-effector pose we used an alternative strategy, where three AR Tags were placed on the end-effector, and the ALVAR package [50] was employed to track the tags and precisely estimate the wrist's 3D pose. As we are only modifying the robot and not the environment, this method does not limit the applicability of the solution, while providing more stability and accuracy compared to the previous approach.

When any wrist tag is visible to the robot, the ALVAR package outputs the pose of the main tag M. To obtain the pose of the hand h, which is the point that should approach the object, as shown in Figure 5, we establish a transform t_h^m that describes the hand in the marker frame. A simple frame transformation yields the required hand pose in camera frame: $\hat{h}^c = \hat{M}_m^c t_h^m$.

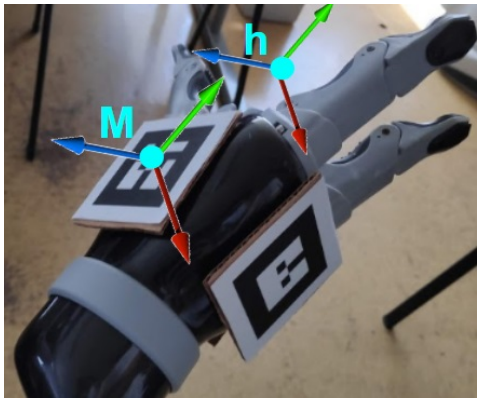


Figure 5. End-effector localization through ALVAR tags (M: marker pose, h: hand pose).

3.4.3. Visual-Servo Control

After estimating the object ($\hat{g}^c$) and end-effector ($\hat{h}^c$) poses in the camera-frame, the error is computed as the difference between the two and minimized by the following proportional control law [37]:

$$u^0 = -k\, \hat{T}_c^0(\hat{h}^c - \hat{g}^c) \tag{1}$$

where $\hat{T}_c^0$ is the transformation matrix from camera to root frame, based on measurements; k is a proportional gain parameter; and u^0 is the resulting root-frame velocity required for the arm to approach the object.

Since both poses are subject to the same camera calibration, the accuracy of the system is independent from calibration errors. The system is also robust to joint sensor errors as the end-effector is being continuously observed in the camera frame. To control the arm in joint-velocity mode, the end-effector's Cartesian velocity must be translated into six joint velocities for the arm's revolute joint actuators. This is done through *inverse differential kinematics* where the arm's Jacobian matrix is inverted through singular value decomposition and multiplied by the end-effector velocity vector, resulting in a column vector of joint velocities, sent to the arm driver. Figure 6 shows the diagram for the described architecture.

Visual servoing is used on the final grasp approach, after the end-effector is already visible to the MBot's head camera. We developed a complete grasping pipeline, making use of the MoveIt! ROS framework, that begins by placing the arm in a pregrasp pose (selected based on a modular criterion, currently the object's height), which subsequently activates visual servoing, stopping when an acceptable distance s (configurable, currently 1.2 cm) is reached, at which point the fingers are closed and the object is grasped.

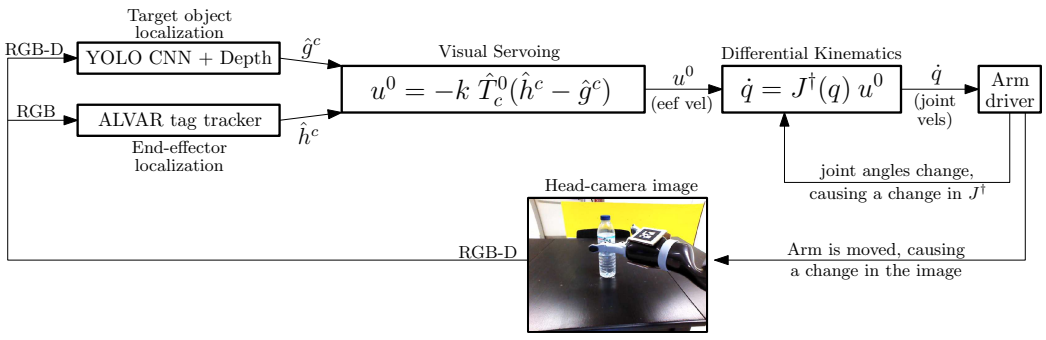


Figure 6. Diagram of the visual servoing architecture for reaching and grasping objects.

3.4.4. Pick and Place Pipeline

The complete Pick and Place pipeline is implemented as a state machine, and its pseudo-code is presented in Algorithm 3:

Algorithm 3: Object-grasping pipeline

```
Function PickPlace(object_type, target_pos):
    start_localizer(object_type)
    sweep_head()
    turn_head_to_object(object_tf)
    pose ← select_pregrasp_pose(object_tf)
    pregrasp(pose)
    visual_servo()
    close_gripper()
    lift_eef()
    move_eef(target_pos)
    lower_eef()
    open_gripper()
    go_to_pose('mbot_resting')
```

The pipeline is triggered by calling an ROS service, using two fields as input: the type of object to grasp and the target pose where the object should be placed. The position of the object is tracked by the YOLO CNN and depth estimator, and its position (*object_tf*) is published as an ROS tf transform. The `sweep_head` procedure turns mbot's head left and right to look for the object and allows it to build an octomap of the scene, which is used for collision avoidance in the arm's motion planning. The head is then turned so that the object is centered in the frame. The `select_pregrasp_pose` procedure is then executed, with the goal of computing an end-effector pose that positions the wrist in the frame, enabling the use of visual servoing. The MoveIt! framework is used to calculate and execute the pregrasp motion plan. With the end-effector in frame, the final approach is performed through visual servoing, minimizing the error in the camera frame until it reaches a threshold, at which point the gripper is closed, and the object is lifted. The end-effector is moved to place the object on the target position relative to the table, and it is then lowered, the gripper is opened and the arm returns to the resting position. Figure 7 shows the robot executing the pipeline, and more examples can be seen in a video available at https://youtu.be/CZaLNTZ_ITU (accessed on 30 October 2020).

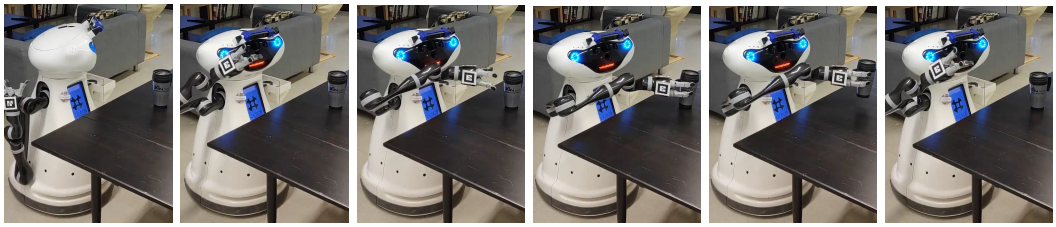


Figure 7. Successful Pick and Place execution by the Mbot robot.

4. Functionality Benchmark

This section describes the functionality benchmarks (FBMs) and the automatic benchmarking system that was designed to automatically evaluate the performance of the three robot functionalities addressed in this paper. These three functionality benchmarks will be part of the ERL benchmarking system that evaluates domestic robot solutions at different certified testbeds around Europe.

4.1. Automatic Benchmarking System

The benchmarking system employs a software called Referee, Scoring and Benchmarking Box (RSBB), which was developed to support and execute benchmarks. The objective of the system is, first, to measure and evaluate the behavior of robots and, second, to collect the benchmark information that can later be analyzed and made available as datasets. To achieve this, the RSBB wirelessly interacts with the robots, interacts with users through a

graphical user interface (GUI) and with other systems such as home automated devices and Motion Capture (MoCap) systems to obtain the ground-truth information of robots, objects and people.

The RSBB is a collection of ROS packages that communicate with each other through ROS topics and services and with the robots through the Protobuf protocol (see Figure 8). The RefBox (Referee Box) acts as the core for the communication between the robot, the GUI (Graphical User Interface), the BmBox (Benchmarking Box) and other external systems such as the MoCap system and Home Automation Devices. The BmBox loads and executes the benchmark scripts when requested by the RefBox.

The RSBB executes a benchmark trial by running a script that sends the goals and benchmark related information to the robots, collects the results and the ground-truth information about the environment and finally computes the score for the trial. The Benchmark Scripts are implemented on top of a framework that facilitates the interaction with the core in order to exchange commands and information from the robots and people, in particular the referee, who controls the RSBB during the execution of the benchmark. The benchmark scripts also communicate with the external software to collect additional information, e.g., the MoCap software. The robots are connect to the RSBB on a Wireless network (see Figure 9). The robots communicate with the RSBB through a ROS package running on the robot. This package provides a simplified interface exposing the relevant topics and services for each specific benchmark and makes the communication protocol transparent to the users. To isolate the ground truth data from the robots, parts of the system run on different networks than the one dedicated to the communication between robot and the RSBB. Hence, the position measured from the MoCap system is not accessible by the robots.

The RSBB automatically executes all the packages needed to execute a benchmark and the user only selects the benchmark to execute and the robot to connect to, allowing any person to easily install and operate the RSBB. Information and instructions from the benchmark script are shown in a Graphical User Interface (GUI) during the execution to guide the user (see Figure 10).

Another objective of the RSBB is to facilitate scripting new benchmarks. A benchmark script is composed of a sequence of goals for the robot and a set of manual operations presented to the user. Goals are transmitted through an ROS service and are used to send information about the goal (e.g., the coordinates where an object should be placed, see Section 4.4) and to receive its execution result (e.g., which person the robot recognized, see Section 4.3). A manual operation provides instructions or requests information from the user (e.g., to place the black cup on the table or ask Peter to move inside the target area). The scripting framework also provides a simplified interface to score and completely log the benchmarks.

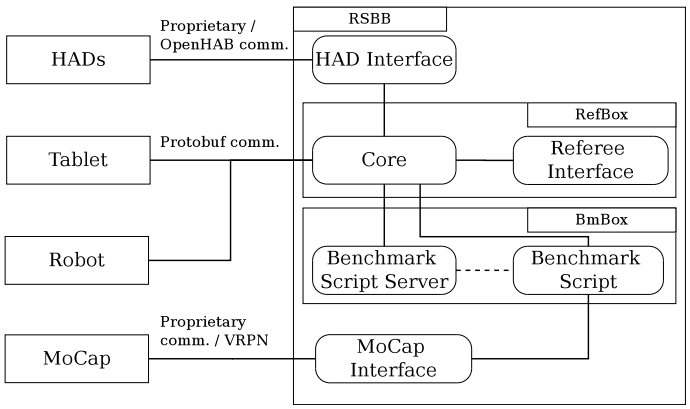


Figure 8. RSBB packages and communication diagram.

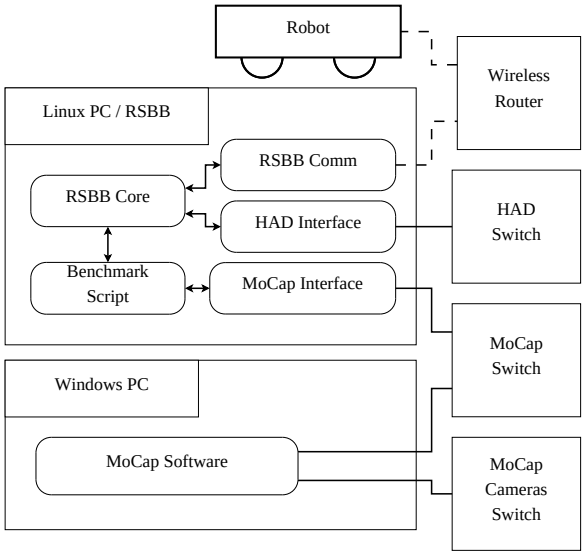


Figure 9. RSBB machine and its network diagram. Solid lines indicate wired connection, while dashed lines indicate wireless connection.

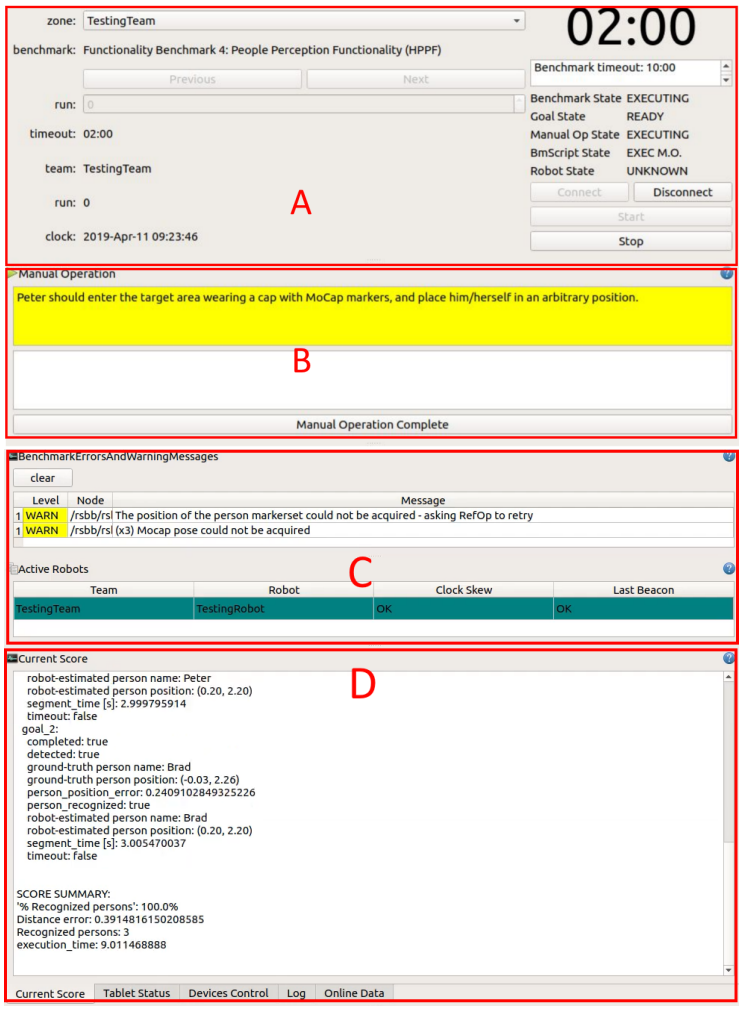


Figure 10. Panels of the RSBB GUI: (**A**) Benchmark control: To select the target robot and benchmark, and to start/stop the benchmark; (**B**) Manual operation: Prints a desired manual action to be executed by the operator, who clicks on the button when it is done; (**C**) Benchmark monitoring: Shows errors/warnings and the currently active robots; (**D**) Score: Displays the current benchmark score.

The RSBB software, along with its documentation and tutorials, is now publicly available at (RSBB: https://github.com/rockin-robot-challenge/rsbb (accessed on 10 August 2020), RSBB robot communication package: https://github.com/rockin-robot-challenge/at_home_rsbb_comm_ros (accessed on 10 August 2020), Benchmark script tutorial: https://github.com/rockin-robot-challenge/rsbb/blob/master/rsbb_etc/doc/bmbox/benchmark_script_tutorial.md (accessed on 10 August 2020)).

4.2. People Perception

A functionality benchmark for evaluating the ability of robots in correctly detecting, identifying and locating humans was designed and developed. The benchmark requires an empty space of predefined dimensions (3 m × 3 m) that will host a set of human targets. The coordinates of this area and the world's reference frame are provided to the the benchmark users. This area is captured by a MoCap system that measures the ground-truth position of the humans inside the target area. Human targets are asked to enter the target area, upon request of the RSBB system, while holding a MoCap marker set, shown in Figure 11, or to wear a hat with the markers.

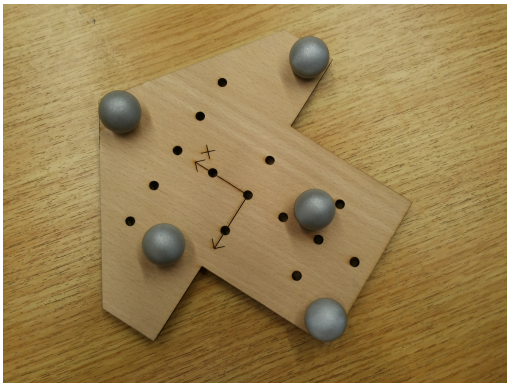


Figure 11. MoCap marker set used on the human targets and the robots to measure their true position in the world.

A set of human targets consisting of both males and females are chosen for the benchmark. The procedure for this FBM is described as follows:

1. The robot is placed on a predefined location outside the specified target and is connected to the RSBB network.
2. The test starts by the RSBB upon receiving the readiness signal from the robot.
3. The RSBB then randomly selects a subject from the set of human targets and requests the person to move inside the area.
4. After the person is in place, the operator uses the RSBB's GUI to press the "Manual Operation Complete" button.
5. The RSBB sends a request signal for the robot to start with the the functionality attempt, i.e., to detect, identify and locate the target.
6. The robot then communicates the perception results to the RSBB.
7. The RSBB asks the subject to move out of the area and randomly selects the next person to enter.

Steps 3 to 5 are repeated for all subjects, and the RSBB automatically computes and communicates the trial score.

For each perception attempt, the robot is expected to communicate the following:

- The 2D position of the person with respect to the world reference frame.
- The identity of the person who is inside the target area.

At the end of a trial, the RSBB evaluates the perception performance of a robot by computing:

1. The percentage of correctly recognized peoples;

2. The localization error for all the detected peoples;
3. The trial's execution time.

Before the benchmark is executed, the robot can collect information about the subjects in order to collect training data.

4.3. Person Following

This FBM was designed to evaluate the capabilities of robots in effectively following humans.

The steps for the procedure of the FBM are as follows:

1. The robot is placed on the starting position in front of a person.
2. The test starts by the RSBB, upon receiving the readiness signal from the robot.
3. A start signal is sent by the RSBB, and the benchmark starts.
4. The person starts to walk and visits a set of locations inside the arena, occasionally stopping and then resuming the walking.
5. The robot is expected to maintain a desired distance with the person while avoiding obstacles and other people that can interrupt the motion of the robot.
6. The benchmark is terminated after a predefined duration, and a timeout signal is sent out by the RSBB.

The performance of the following behavior is computed with the help of the MoCap system that continuously measures the true pose of the robot and the target at all times. Both the robot and the target person are equipped with MoCap markers. To derive the pose of the robot's odometric center with respect to the origin, the relative pose of the marker set with respect to the robot's center is required. This is simply obtained by the RSBB before the benchmark by placing the robot on the origin of the test bed and measuring the transformation between the marker set and the origin. The three-dimensional poses of the human target and the robot are projected onto the ground plane, and their Euclidean ground distance is continuously computed.

A performance score is computed based on the following:

1. M_A: The accuracy in following and maintaining the desired relative distance to the target;
2. M_T: The total covered distance while following the target. The robot is considered to be following the target if it is within a tolerance range of the desired distance.

The previous criteria encourage both accurate and fast solutions with minimum interruption requests by the robot. Since this benchmark requires continuous tracking of the robot and the person by the MoCap system, which might be interrupted due to marker occlusions or unexpected movements in regions not captured by the MoCap system, a *benchmark reliability* metric is also calculated indicating how well the benchmark system was able to capture the trial.

The three metrics are described in the following equations:

$$M_A = \frac{\sum_{s \in S} |D(\text{robot}_s, \text{person}_s) - D_{desired}|}{\#S} \tag{2}$$

$$M_T = \sum_{s \in S} D(\text{r}_{s-1}, \text{r}_s), \quad \text{if } D_{min} \leq D(\text{r}_s, \text{p}_s) \leq D_{max} \tag{3}$$

$$\textit{Benchmark Reliability} = \frac{\#S}{\#S + \#F} \tag{4}$$

where S and F are the sets of successful and failed MoCap samples, respectively; the $D(a, b)$ function is the L2 norm between two 2D positions; and D_{min}, D_{max} and $D_{desired}$ are configurable parameters (for the experiments in this work, their values are 0.15 m, 3.5 m and 2 m, respectively).

4.4. Pick and Placing

This FBM evaluates the ability of a domestic service robot to correctly grasp, pick and place objects. In particular, the benchmark focuses on the picking and placing capabilities of robots, which is an important functionality for domestic applications, such as to prepare a dining table.

Figure 12 illustrates a schematic diagram of the benchmarking procedure. The RSBB requests the operator to place objects one by one, on top of a table in front of the robot. The robot needs to correctly grasp, lift up and then place the object on a target position that is specified by the RSBB.

The steps for the FBM are as follows:

1. The robot is placed in front of the table.
2. The benchmark trial starts by the RSBB, after the readiness signal is received by the robot.
3. The RSBB randomly selects an item from the list of items and asks the referee to place it on the table in a random location.
4. The RSBB sends the execution request with information about the identity of the object and the desired target location.
5. The robot must then grasp, pick up and place the object on the specified target location.
6. The robot sends the complete attempt signal to the RSBB and announces its readiness for the next object.

This process is repeated for all items in the item list. Evaluation of the performance is based on the following:

1. The percentage of correctly grasped objects. A successful grasp is automatically detected by the RSBB system once it detects that the height of the object has increased by a pre-defined threshold;
2. The error in the placement position with respect to the specified target location;
3. The execution time.

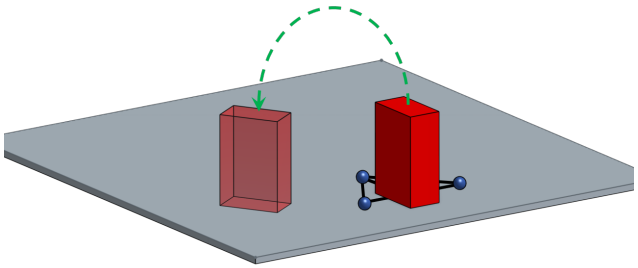


Figure 12. Illustration of the Pick and Placing benchmark with the objective of grasping, picking and placing the object on a specified target position. MoCap markers attached under the object allow the MoCap system to continuously measure the true object pose.

A set of common household objects of different shapes is currently used for this benchmark. As the focus of this FBM is on picking and placing capabilities, the nature of the object placed on the table is communicated to the robot by the RSBB to assist the robot with perception. In the future, we plan to release a set of standard 3D printed objects for this benchmark to further facilitate perception and to obtain a detailed performance evaluation covering a wide range of object shapes and rigidities.

To acquire the true pose of an object, a stand equipped with MoCap markers is placed under the objects on the table. An L-shaped tool is also attached to the table, carrying both MoCap markers and a visual tag that indicate the reference of the table to the MoCap system and to the robot. Both tools are shown in Figure 13. This allows the RSBB to acquire the true pose of objects relative to the table origin.

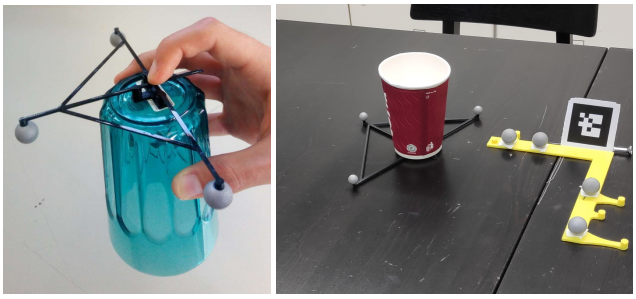


Figure 13. Object stand and table origin tool, both fitted with MoCap markers.

5. Experiments and Results

This section describes evaluation of the functionalities presented in Section 3, implemented and tested on our Mbot domestic service robot, using the Functionality Benchmarks and the automatic benchmarking system presented in Section 4. Videos of trials for each of the functionality benchmarks are available below (People Perception: https://www.youtube.com/watch?v=pcJfcczA964 (accessed on 3 December 2020), People Following: https://www.youtube.com/watch?v=reOBLMX4X5U (accessed on 3 December 2020), Pick and Placing: https://www.youtube.com/watch?v=lv5KLJC40pI (accessed on 3 December 2020)).

5.1. People Perception

The People Perception method presented in Section 3.2 was implemented on our Mbot robot, and the People perception FBM was used to test and evaluate the method. As mentioned before, in the People Perception FBM, the benchmarking system automatically records the true and estimated position of every person, computes the position error and checks to see if the robot has correctly recognized the person. An average position error and a percentage of correct identifications is computed at the end of a trial. Table 1 shows results from a single benchmark trial. Figure 14 illustrates the perception results from six different trials of this FBM.

Table 1. People Perception FBM results for 5 People.

	Position Error	Detection	Time
Person 1	0.1918 m	0%	9.0 s
Person 2	0.0735 m	100%	10.7 s
Person 3	0.1481 m	0%	7.5 s
Person 4	0.1916 m	100%	11.2 s
Person 5	0.2502 m	100%	11.2 s
Trial Result	0.1710 m	60%	9.9 s

Figure 14. People Perception FBM results per trial.

5.2. People Following

In the People Following FBM, the main data collected are the position of the robot and the position of the person from which the evaluation metrics are calculated automatically. Figure 15 shows the path of the robot when following a person inside the ISRoboNet@Home testbed, while using the People Following method described in Section 3.3.

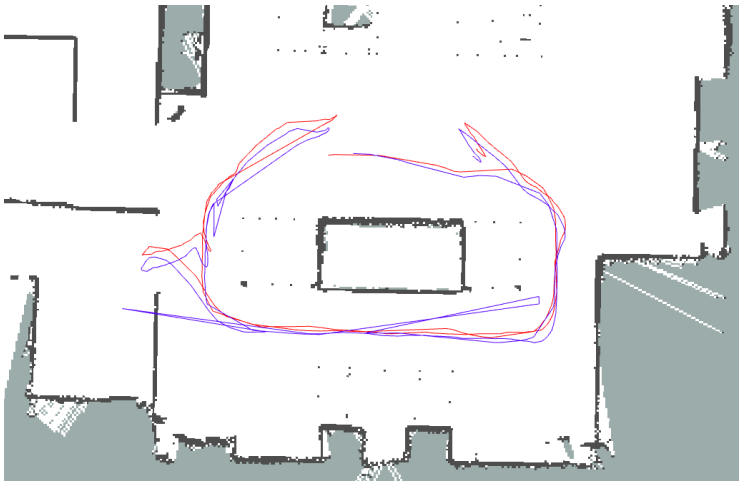


Figure 15. Path of robot, where the red lines indicate the path estimated by the robot and the blue lines indicate the paths measured by the MoCap system.

Figure 16 shows the FBM results from seven different trials, indicating the three output metrics that are provided by the benchmarking system. The total distance covered by the robot while following and the average deviation from the desired distance indicate how well the robot has performed the following behavior. The benchmark reliability metric shows how well the benchmark trial was captured by the benchmarking system in different runs. The reliability score can easily be increased to near 100% by increasing the number of MoCap cameras and carefully positioning the cameras to cover all regions of the arena.

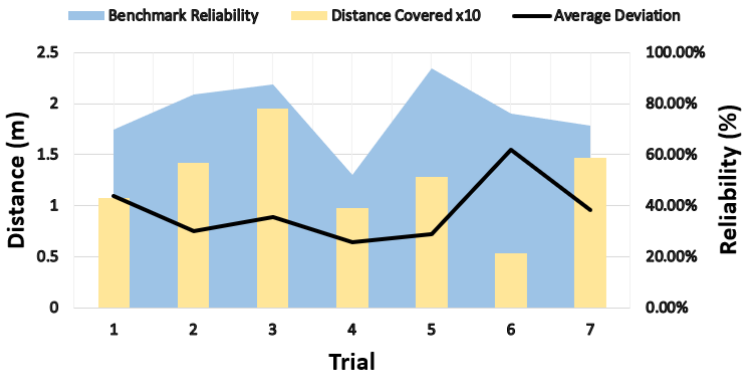


Figure 16. People Following FBM results per trial.

Figure 17 shows the relative distance between the robot and the person as measured by the robot and the MoCap system. The desired following distance to be maintained by the robot was 2 m, and the robot was considered to be following if its relative distance was inside a threshold indicated by the yellow region. The MoCap distance of 0 symbolizes the moments when the MoCap system was unable to locate the person or the robot, and hence that data point was not considered.

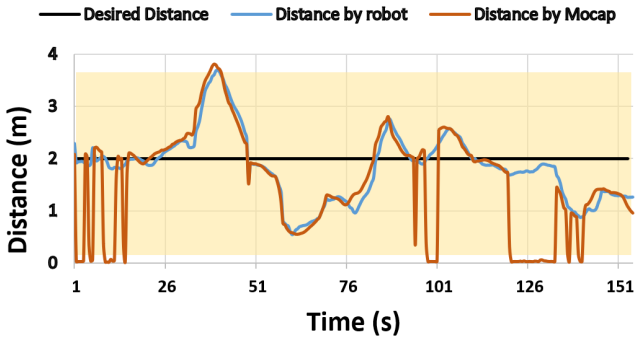


Figure 17. The relative distance between the person and the robot, estimated by the robot and the MoCap system in one of the FBM trials. The threshold for considering the robot to be following a person is indicated by the yellow area.

5.3. Pick and Placing

To evaluate the Pick and Placing method and its functionality benchmark, five objects were chosen and added to the FBM's configuration file: a water bottle, a tall mug, a large coffee cup, an espresso cup, and an orange. Figure 18 shows the objects used for the test.

Figure 18. Household objects used in the Grasping and Manipulation FBM.

The benchmark routine was repeated three times. In each trial, objects were placed on the table one object at a time, randomly selected by the RSBB, for the robot to pick and move to a target location as defined by the RSBB.

Figure 19 displays the grasping rate (the percentage of objects successfully grasped and lifted by at least 2 cm) and the placement accuracy (the average placement error to the object's target position) for the three trials.

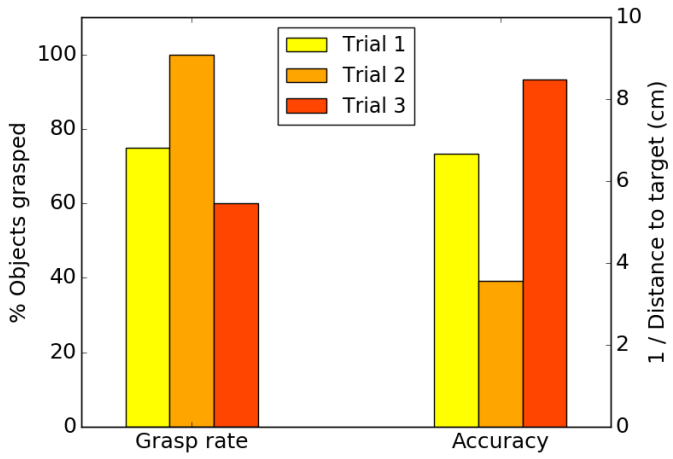


Figure 19. Pick and Placing FBM results per trial.

The RSBB calculates both scores automatically by tracking the 3D pose of the object obtained by the MoCap system. Figure 20 show the vertical and horizontal movement of the object, tracked by the RSBB system, while being grasped and re-positioned as part of a benchmark trial.

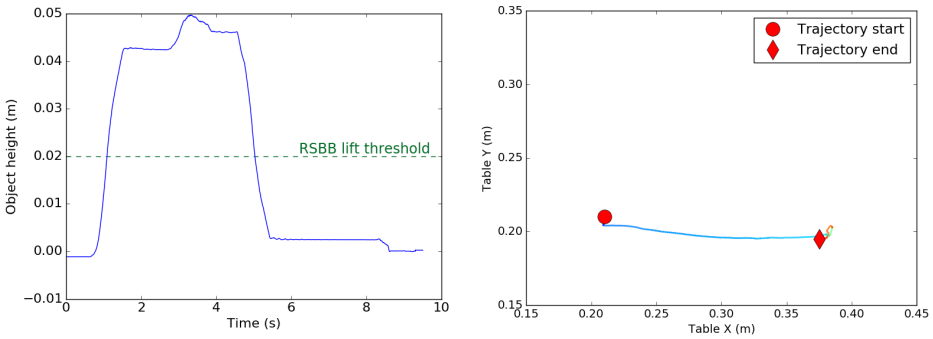


Figure 20. A Pick and Place attempt captured by the RSBB system. Left: Variation in the object's height relative to the table, describing a successful picking. Right: X, Y trajectory of the object on the table, showing a correct re-positioning of the object to the target location.

6. Conclusions

This paper described efforts toward the development of three main functionalities for a mobile service robots aimed to perform daily domestic tasks, with the goal of providing assistance to elderly or handicapped persons inside a house. Furthermore, a set of novel functionality benchmarks and an innovative performance evaluation system was described for the systematic evaluation of the three robot functionalities. These benchmarks and the benchmarking system are now available to be used at different certified test beds around Europe under the European Robotics League (ERL)-Consumer that evaluate advanced domestic robot solutions under the umbrella of scientific robot competitions. Currently, a total of six functionality benchmarks have been implemented and are ready to be used with the proposed automatic benchmarking system. We aim to continue this route to include other FBMs and cover other functionalities needed for a domestic service robot. Furthermore, we aim to enhance the existing FBMs to include additional benchmarking metrics and algorithms that allow automatic calculation of performance scores and to further increase the benchmark autonomy. Equipping the testbed and objects with sensors to detect and evaluate robot–object interactions (for example, to quantitatively characterize the capability of a robot in interacting with objects, opening doors or pushing a wheel chair) is among the future work envisioned for this project.

Author Contributions: M.B. designed the functionality benchmarks and supervised their implementation, provided theoretical support and wrote the manuscript. J.P. developed the benchmark scripts, designed and developed the Pick and Placing functionality of the robot and contributed in the writing of the manuscript. R.B. implemented the robot's state machine and the people detection and localization modules, executed the experiments and contributed in the writing of the manuscript. E.P. developed the RSBB's main software. E.F. and C.A. implemented the person recognition module and assisted with the execution of the experiments. P.L. provided theoretical support and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by EU Horizon 2020 Program for research, technological development and demonstration under grant agreement no. 780086 and is partially supported by ISR/LARSyS Strategic Funding through the FCT project UIDB/50009/2020.

Institutional Review Board Statement: Not applicable. Ethical review and approval were waived for this study due to the internal regulations of the university of Lisbon and since all experiments were low risk and only involved volunteers from the research team.

Informed Consent Statement: This article does not contain any studies with animals performed by any of the authors. Only volunteers from the research team participated in the human-robot interactions. Informed consent was obtained from all individual participants.

Data Availability Statement: The RSBB software and documentation, as well as a tutorial explaining how to write a benchmark, is now publicly available at: https://github.com/rockin-robot-challenge/rsbb (accessed on 15 August 2020). The RSBB communication package for the robots is available at: https://github.com/rockin-robot-challenge/at_home_rsbb_comm_ros (accessed on 15 August 2020). All other codes and data related to the implemented robot functionalities are available from the corresponding author by request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wise, M.; Ferguson, M.; King, D.; Diehr, E.; Dymesich, D. Fetch and freight: Standard platforms for service robot applications. In Proceedings of the Workshop on Autonomous Mobile Service Robots, New York, NY, USA, 11 July 2016.
2. Liu, J.; Wang, Y.; Li, B.; Ma, S. Current research, key performances and future development of search and rescue robots. *Front. Mech. Eng. China* **2007**, *2*, 404–416. [CrossRef]
3. Bai, L.; Guan, J.; Chen, X.; Hou, J.; Duan, W. An optional passive/active transformable wheel-legged mobility concept for search and rescue robots. *Robot. Auton. Syst.* **2018**, *107*, 145–155. [CrossRef]
4. Basiri, M.; Gonçalves, J.; Rosa, J.; Bettencourt, R.; Vale, A.; Lima, P. A multipurpose mobile manipulator for autonomous firefighting and construction of outdoor structures. *Field Robot.* **2021**, *1*, 102–126. [CrossRef]
5. Siciliano, B.; Khatib, O. (Eds.) Domestic Robotics. In *Springer Handbook of Robotics*; Springer International Publishing: Cham, Switzerland, 2016; pp. 1729–1758. [CrossRef]
6. Sinapov, J.; Stoytchev, A. Object category recognition by a humanoid robot using behavior-grounded relational learning. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 184–190. [CrossRef]
7. Young, J.E.; Hawkins, R.; Sharlin, E.; Igarashi, T. Toward acceptable domestic robots: Applying insights from social psychology. *Int. J. Soc. Robot.* **2009**, *1*, 95–108. [CrossRef]
8. Müller, A.C.; Behnke, S. Learning depth-sensitive conditional random fields for semantic segmentation of RGB-D images. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6232–6237. [CrossRef]
9. Messias, J.; Ventura, R.; Lima, P.; Sequeira, J.; Alvito, P.; Marques, C.; Carriço, P. A robotic platform for edutainment activities in a pediatric hospital. In Proceedings of the 2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Espinho, Portugal, 14–15 May 2014; pp. 193–198. [CrossRef]
10. Muszynski, S.; Stückler, J.; Behnke, S. Adjustable autonomy for mobile teleoperation of personal service robots. In Proceedings of the 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication, Paris, France, 9–13 September 2012; pp. 933–940. [CrossRef]
11. Gu, S.; Holly, E.; Lillicrap, T.; Levine, S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3389–3396. [CrossRef]
12. Pineda, L.A.; Salinas, L.; Meza, I.V.; Rascon, C.; Fuentes, G. Sitlog: A programming language for service robot tasks. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 358. [CrossRef]
13. Matamoros, M.; Seib, V.; Memmesheimer, R.; Paulus, D. RoboCup@Home: Summarizing achievements in over eleven years of competition. In Proceedings of the 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Torres Vedras, Portugal, 25–27 April 2018; pp. 186–191. [CrossRef]
14. Amigoni, F.; Bastianelli, E.; Berghofer, J.; Bonarini, A.; Fontana, G.; Hochgeschwender, N.; Iocchi, L.; Kraetzschmar, G.; Lima, P.; Matteucci, M.; et al. Competitions for Benchmarking: Task and Functionality Scoring Complete Performance Assessment. *IEEE Robot. Autom. Mag.* **2015**, *22*, 53–61. [CrossRef]
15. European Robotics League. Available online: https://www.eu-robotics.net/robotics_league/ (accessed on 13 October 2020).
16. Basiri, M.; Piazza, E.; Matteucci, M.; Lima, P. Benchmarking Functionalities of Domestic Service Robots Through Scientific Competitions. *KI-Künstliche Intell.* **2019**, *33*, 357–367. [CrossRef]
17. Stuckler, J.; Holz, D.; Behnke, S. RoboCup@Home: Demonstrating Everyday Manipulation Skills in RoboCup@Home. *IEEE Robot. Autom. Mag.* **2012**, *19*, 34–42. [CrossRef]
18. Cartucho, J.; Ventura, R.; Veloso, M. Robust object recognition through symbiotic deep learning in mobile robots. In Proceedings of the 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 2336–2341.
19. RoCKIn: Robot Competitions Kick Innovation in Cognitive Systems. Available online: http://rockinrobotchallenge.eu (accessed on 13 December 2018).

20. Lima, P.U. The RoCKIn Project. In *RoCKIn: Benchmarking Through Robot Competitions*; IntechOpen: London, UK, 2017; Chapter 2, p. 765. [CrossRef]
21. Li, D.; Li, L.; Li, Y.; Yang, F.; Zuo, X. A Multi-Type Features Method for Leg Detection in 2-D Laser Range Data. *IEEE Sens. J.* **2018**, *18*, 1675–1684. [CrossRef]
22. Weinrich, C.; Wengefeld, T.; Schroeter, C.; Gross, H. People detection and distinction of their walking aids in 2D laser range data based on generic distance-invariant features. In Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication, Edinburgh, UK, 25–29 August 2014; pp. 767–773. [CrossRef]
23. Davis, J.W.; Keck, M.A. A Two-Stage Template Approach to Person Detection in Thermal Imagery. In Proceedings of the 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)—Volume 1, Breckenridge, CO, USA, 5–7 January 2005; Volume 1, pp. 364–369. [CrossRef]
24. Nguyen, D.T.; Li, W.; Ogunbona, P.O. Human detection from images and videos: A survey. *Pattern Recognit.* **2016**, *51*, 148–175. [CrossRef]
25. Shotton, J.; Girshick, R.; Fitzgibbon, A.; Sharp, T.; Cook, M.; Finocchio, M.; Moore, R.; Kohli, P.; Criminisi, A.; Kipman, A.; et al. Efficient Human Pose Estimation from Single Depth Images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2821–2840. [CrossRef] [PubMed]
26. Muñoz-Salinas, R.; Aguirre, E.; García-Silvente, M. People detection and tracking using stereo vision and color. *Image Vis. Comput.* **2007**, *25*, 995–1007. [CrossRef]
27. Martinel, N.; Luca Foresti, G.; Micheloni, C. Aggregating deep pyramidal representations for person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.
28. Zhu, Z.; Jiang, X.; Zheng, F.; Guo, X.; Huang, F.; Sun, X.; Zheng, W. Viewpoint-Aware Loss with Angular Regularization for Person Re-Identification. *AAAI Conf. Artif. Intell.* **2020**, *34*, 13114–13121. [CrossRef]
29. Haque, A.; Alahi, A.; Li, F.-F. Recurrent Attention Models for Depth-Based Person Identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
30. Ye, M.; Shen, J.; Crandall, D.; Shao, L.; Luo, J. Dynamic Dual-Attentive Aggregation Learning for Visible-Infrared Person Re-identification. In *Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 229–247.
31. Gupta, M.; Kumar, S.; Behera, L.; Subramanian, V.K. A Novel Vision-Based Tracking Algorithm for a Human-Following Mobile Robot. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1415–1427. [CrossRef]
32. Xing, G.; Tian, S.; Sun, H.; Liu, W.; Liu, H. People-following system design for mobile robots using kinect sensor. In Proceedings of the 2013 25th Chinese Control and Decision Conference (CCDC), Guiyang, China, 25–27 May 2013; pp. 3190–3194. [CrossRef]
33. Susperregi, L.; Martínez-Otzeta, J.M.; Ansuategui, A.; Ibarguren, A.; Sierra, B. RGB-D, laser and thermal sensor fusion for people following in a mobile robot. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 271. [CrossRef]
34. Basiri, M.; Gonçalves, J.; Rosa, J.; Vale, A.; Lima, P. An autonomous mobile manipulator to build outdoor structures consisting of heterogeneous brick patterns. *SN Appl. Sci.* **2021**, *3*, 558. [CrossRef]
35. Kumar, R.; Lal, S.; Kumar, S.; Chand, P. Object detection and recognition for a pick and place robot. In Proceedings of the Asia-Pacific World Congress on Computer Science and Engineering, Nadi, Fiji, 4–5 November 2014; pp. 1–7.
36. Lozano-Pérez, T.; Jones, J.L.; Mazer, E.; O'Donnell, P.A. Task-level planning of pick-and-place robot motions. *Computer* **1989**, *22*, 21–29. [CrossRef]
37. Hutchinson, S.; Hager, G.D.; Corke, P.I. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.* **1996**, *12*, 651–670. [CrossRef]
38. Collewet, C.; Marchand, E. Photometric Visual Servoing. *IEEE Trans. Robot.* **2011**, *27*, 828–834. [CrossRef]
39. Bateux, Q.; Marchand, E.; Leitner, J.; Chaumette, F.; Corke, P. Training Deep Neural Networks for Visual Servoing. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3307–3314. [CrossRef]
40. Lenz, I.; Lee, H.; Saxena, A. Deep learning for detecting robotic grasps. *Int. J. Robot. Res.* **2015**, *34*, 705–724. [CrossRef]
41. Levine, S.; Finn, C.; Darrell, T.; Abbeel, P. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* **2016**, *17*, 1334–1373.
42. Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; et al. Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. In Proceedings of the 2018 Conference on Robot Learning, Zurich, Switzerland, 29–31 October 2018; pp. 651–673.
43. Ventura, R.; Basiri, M.; Mateus, A.; Garcia, J.; Miraldo, P.; Santos, P.; Lima, P. A domestic assistive robot developed through robot competitions. In *Ijcai 2016 Workshop on Autonomous Mobile Service Robots*; Intelligent Robots and Systems Group (IRSg): New York, NY, USA, 2016.
44. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
45. Lima, P.U.; Azevedo, C.; Brzozowska, E.; Cartucho, J.; Dias, T.J.; Gonçalves, J.; Kinarullathil, M.; Lawless, G.; Lima, O.; Luz, R.; et al. SocRob@ Home. *KI-Künstliche Intell.* **2019**, *33*, 343–356. [CrossRef]
46. Geitgey, A. Face Recognition. Available online: https://github.com/ageitgey/face_recognition (accessed on 10 September 2020).

47. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. [CrossRef]
48. Kazemi, V.; Sullivan, J. One Millisecond Face Alignment with an Ensemble of Regression Trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
49. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
50. ALVAR. Library for Virtual and Augmented Reality. Available online: http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/ (accessed on 5 November 2020).