Article

Integration of shared autonomy system for grasping multiple objects in the domestic environment

Jaeseok Kim ¹, Raffaele Limosani ¹ and Filippo Cavallo ^{1,*}

¹ BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, Italy.; jaeseok.kim@santannapisa.it, raffaele.limosani@santannapisa.it

* Correspondence: filippo.cavallo@santannapisa.it; Tel.: +39 050 883474

Academic Editor: name

Version October 3, 2018 submitted to Journal Not Specified

Abstract: In recent years, autonomous robots have proven capable of solving tasks in complex environments. In particular, robot manipulations in activities of daily living (ADL) for service robots have been in wide use. However, manipulations such as loading a dishwasher or folding laundry are difficult to automate robustly. In addition, manipulations of grasping multiple objects in domestic environments present difficulty. To perform those applications better, we developed robotic systems based on shared autonomy by combining the cognitive skills of a human operator with autonomous robot behaviors. In this work, we present techniques for integration of a shared autonomy system for assistive mobile manipulation and new strategies to support users in the domestic environment. We demonstrate that the robot can grasp multiple objects with random size at known and unknown table heights. Specifically, we developed three strategies for manipulation. From experimental results, we observed that the first strategy has the highest success rate (70% for random objects) up to 70 cm table height. Two other strategies perform better for 80 cm to 100 cm table height. The success rate of the second and third strategies shows an average 63.3% and 73.3%, respectively, for grasping random objects. We also demonstrated these strategies using two intuitive interfaces, a visual interface in rviz and a voice user interface with speech recognition, which are suitable for elderly people. In addition, the robot can select strategies automatically in random scenarios, which make the robot intelligent and able to make decisions independently in the environment. We obtained interesting results showing that the robot adapts to the environmental variation automatically. After these experimental demonstrations, our robot shows the capabilities for employment in domestic environments to perform actual tasks.

Keywords: Mobile manipulation; Shared autonomy; Assistive robotics; Grasping multiple objects; Service robotics

1. Introduction

In the home environment, perception is used to recognize a variety of objects; however, a service robot might not be able to detect all of the objects in every circumstance. In other words, when the robot encounters multiple objects on a table, it needs large computation time; otherwise, the robot might fail to accomplish the given tasks. However, if a human can support the judgment of the robot, the robot can acquire the specific object needed quite easily. Furthermore, the calculation time for classifying and selecting the appropriate object can be reduced significantly.

Elderly people tend to spend more time at home and need care in the home due to declining capabilities and increasing illnesses. To support the situation, an intelligent system, which can assist elderly people, is necessary [1]. One of the options is to develop a service robot, which assists in human

Figure 1. The domestic robot Doro, which consists of depth and stereo cameras, robotic arm and mobile-base platform, is useful manipulation task in domestic environment.

activities with smart home technology that connects between the human and the robot. For this reason, many studies researching service robots are underway [2,3].

To find solutions for performing tasks in the home environment, many types of service robots have been developed. In particular, robots that provide shared autonomy and activities of daily living (ADL) for older people have been developed. One representative service robot is Care-o-bot, which was developed with basic technologies for delivery, navigation, and monitoring for users [4]. Moreover, in recent years, several projects have featured different robots that integrate smart home technology for health-care, shopping, garaging [5], and communication with users by gesture and speech [6].

Despite enhanced functionalities of the service robots, we still face several challenges of ADL in the domestic environment. Particularly with tasks such as grasping objects iteratively in the environments, the capabilities of current robots are still lacking. To solve these problems, robots typically focus on either a fully autonomous or a fully teleoperated system. However, many limitations with perception and manipulation remain. A possible solution to overcome these issues is to use a shared autonomy system, in which the human operator controls the robot in a remote site with a high level of abstraction.

In this paper, we developed integration of a shared autonomy system for grasping multiple objects in the domestic environment. To develop the system, we present a new method that is composed of three mobile manipulation strategies that were operated by the Doro robotic platform (Fig. 1). We focused on development of the strategies for grasping unreachable objects with variable known table height. Also, grasping an object on the unknown table heights was considered using the strategies.

The paper is organized as follows. In the next section, relevant works related to the mobile manipulation tasks for grasping objects and user-interfaces are summarized. In section 3, the system architecture for grasping multiple objects is explained. In section 4, our shared autonomy system is described briefly. In section 5, we discuss implementation of the shared autonomy system, which includes multi-object segmentation, user interfaces, and mobile manipulation strategies. In sections 6 and 7, the experimental setup and experimental results are described. Section 8 contains conclusions and future work.

2. Related work

Mobile manipulation tasks for grasping an object in the domestic environment have been studied extensively over decades [7–11]. Research has been conducted on tasks that are performed with several

Figure 2. The mobile manipulation task is operated by human capabilities applied with Sheridan's four-stage model (a) Information acquisition, (b) Information analysis, (c) Detection and action selection, and (d) action implementation

types of systems, including autonomous manipulation, human-in-the-loop, shared autonomy, and human-robot interaction systems.

The autonomous manipulation system is frequently used to grasp objects in the domestic environment. In particular, the system has been developed for unreachable objects in the environment. The pushing operation system is one of the solutions to manipulate objects. Dogar *et al.* [7] suggest a framework to generate a sequence of pushing actions to manipulate a target object. Also, Kitaev *et al.* [8] present a method of push-grasping in which the robot arm shifts unavoidable objects to grasp a target object in simulation. However, the pushing action system needs adequate space in which to shift or remove objects. In addition, the presence of only one grasp pose in the system does not facilitate grasping objects of various shapes. Moreover, a sequence of manipulation actions to grasp the objects is suggested. Stilman *et al.* [9] use a sampling-based planner to take away the blocking objects in a simulation. For example, to grasp target objects, the robot arm opens a cabinet and removes an object in front of the target object sequentially. Another approach presents a picking and placing or sweeping movement to remove each object around the target object [10]. Moreover, Fromm *et al.* [11] propose a method to plan strategies for a sequence of manipulation actions. The authors did not set a target object, but they considered grasping all objects in a manipulation sequence selected by a search tree method. The sequence of manipulation actions described above show similar human behavior. Therefore, we also consider a sequence of manipulation actions for grasping unreachable objects. However, the discussed literature show that the robot already knows the target object before the robot manipulation starts. To overcome the object selection during the robot manipulation, user-interfaces operated by a shared autonomy system are developed.

Many prior works address development of user-interfaces with shared autonomy systems for object selection. For remote selection of an object by people at home, the graphical point-click interface system was developed [12,13]. The interface is allowed to drag, translate, and rotate to select a target object by a person. In addition, the interface is used to generate waypoints for desired gripper position to conduct grasping tasks [14]. Moreover, the interface supports the grasp point on the object, sets the approaching angle, and adjusts the grasp pose to execute robot motion [15]. In addition, object-centered robot actions operated by service robot have been developed using a tablet PC [16]. Also, a laser point interface is employed for users with tremors and upper body impairment [17]. The interface systems support the object selection problem using human capabilities. However, the interface systems on the papers only consider grasping reachable objects, which are not occluded, and simple task planning is applied. In addition, to select the target object, a human operator should concentrate on the visual display and take their time when selecting an object.

Also, the sequence of manipulation actions for grasping an unreachable object using the supervision interface was developed [18]. The authors used an interface that only supports robot arm control to perform the pushing operation. However, the object selection system was not under consideration. Therefore, to grasp unreachable objects, we developed three mobile manipulation strategies with high-level automation and different grasp poses to support grasp planning. Of course, the object selection system was implemented using a voice and a visual interface for supporting robot perception to build people's ADL at home.

3. System architecture

The goal of our work is to develop a robotic system that will be able to help people in ADL. In particular, we studied the scenario in which a user needs a particular object located on a table and asks the assistant robot to find and bring the object. To successfully perform the task, the assistant robot needs a high level of autonomy and the capability to interact with humans.

In fact, the preferred way to achieve the scenario is to operate the robotic system automatically. Moreover, many different robotic systems have been developed to perform the task without human support. In particular, the robot with a multi-sensory navigation system for the indoor environment supports bringing a target object to the user safely. However, the capability to recognize a target object, calculate eligible grasp poses, and generate task planning for complex tasks are still being researched. The mobile manipulation task, which is part of the robotic system, should be automated. Parasuraman *et al.* [19] proposed a model for different levels of automation that provides a framework made up of four classes: (1) information acquisition, (2) information analysis, (3) decision and action selection, and (4) action implementation. In current work, we adapted the same framework in our robotic system (see Fig. 2).

Information acquisition represents the acquisition of raw sensed data, which consists of distance measurements using a camera and laser sensor. In addition, information about the state of the robot and its environment (information about objects) is provided. To manipulate successfully, this function generally can be automated efficiently and robustly (see Fig. 2(a)).

Information analysis involves object segmentation and obstacle recognition. For the autonomous field, one main challenge is to detect and cluster the object-sensed data in the domestic environment because the sensor data can be incomplete and noisy due to occlusions and outliers. In addition, reliable interpretation of images from the camera remains a largely unsolved research problem [20] (see Fig. 2(b)).

Decision and action selection are performed by the human. The human knows where the object is located and can tell the robot the exact position. Furthermore, the human can support objects that the robot should grasp using visual and user voice interfaces. The grasp point of the objects was inferred using the 3D centroid function in our system to support grasp planning (see Fig. 2(c)).

The last function is called **action implementation**; the motion planning system that generates the path to control the robot arm is used. After the robotic arm finishes following the trajectory, the grasping task is executed (see Fig. 2(d)).

4. Shared autonomy system description

Based on Sheridan's four-stage model, which was previously described, *Decision & Action selection* were conducted as a good starting point for the shared autonomy concept. Moreover, in this model, the human operator only supports object selection in the domestic environment. We aimed to develop a system that is operated by minimum human effort. The human operator contributes by interpreting the environment from the camera images and by choosing a low level of automation in the third stage of Sheridan's model.

The final objective of the mobile manipulation task is to pick up an object on a table of unknown height and bring it to a user. In our system, as the robot finds a table, it measures the table height and adjusts its neck angle accordingly. Then, robot detects and segments multiple objects that are

Figure 3. A simple pictorial diagram with variables such as neck angle(N_θ), table height(T_h) and object height(O_h). Look at the text for the variables applied in the equations.

positioned in two rows (front and back). As the robot completes extraction of the objects with several RGB colors, the user selects a row and a target object using voice and visual interfaces. Based on the table height information, and row and object information (height, length, weight, and distance), the robot selects one of three mobile manipulation strategies that we have developed for grasping an object in the back row. The strategies were designed with different grasp poses according to the table heights.

For grasping an object, we followed two scenarios. In the first scenario, the user employed a known table height fixed at 70, 80, 90, or 100 cm. Moreover, in the second scenario, the user employed an unknown table height which is measured by robot itself, and robot can decide empirically which strategies are better for grasping. From several experimental trials, empirical results suggest three strategy modes for better grasping:

$$\begin{cases} 1 & T_h < 77cm \\ 2 & T_h < 87cm \\ 3 & T_h < 100cm \end{cases}$$

where T_h is table height measured by camera. For grasping objects and controlling the arm, we used Point Cloud Library (PCL) [21] and Moveit [22] library.

5. Implementation of shared autonomy system

The goal of the shared autonomy system is to provide support to improve the quality of human life. For example, if a human operator assigns a task to a service robot with a high level of autonomy using a gadget, the human can use their own cognitive skills for selecting objects. Thus, the task performance with the shared autonomy system is more efficient than an autonomous system, which still has difficulty in recognizing objects in the domestic environment. The main contribution of this paper is the development of motion planning with three strategies of manipulation. To implement the strategies of manipulation, several components such as image preprocessing, multi-object segmentation, visual and voice user interfaces, and action planning were applied. The components were organized to perform key roles in performing fundamental ADL for human life.

The domestic environment contains myriad household objects with different shapes and sizes such as bottle, box, and cup, chairs, tables etc. We considered grasping an object from a table of unknown height in the domestic environment, which schematically is shown in Fig. 3 with our robot. Actually, many studies for grasping objects [7–11] were tested using fixed table height and viewpoint. However, if the viewpoint is changed, detection of objects on the table will be difficult by the robot. Therefore, to overcome the difficulty of detecting objects from a different viewpoint, we adjusted the neck angle of the robot based on table height. Before applying the fixed neck angle, the robot needs to find a table. Thus, the initial neck angle was set at the lowest position to find a lower table height. For detecting the table, the random sample consensus (RANSAC) [23] algorithm was used to filter noises from raw data of the environment and was also applied to segment a table by PCL. After the table was segmented, a point (which is calculated by averaging all the coordinate points on the top surface of

Figure 4. The flow chart of the shared autonomy system; It is included with image preprocessing, multi-object segmentation, object selection and action planning with developed strategies.

the table) was extracted from the base frame of the robot (see Fig. 5), and only the z-axis value was used to calculate table height. Then the value was stored for changing the neck angle and choosing the strategies. Next, the neck angle of the robot was adjusted by the interpolation method. To interpolate the neck angle, we set the maximum and minimum range of neck angle and table height. Moreover, the linearly interpolated neck angle helped the robot to detect multiple objects easily.

$$N_{\theta,d} = N_{\theta,min} + (T_h - T_{h,min}) \frac{N_{\theta,max} - N_{\theta,min}}{T_{h,max} - T_{h,min}} \quad (1)$$

where $N_{\theta,d}$ is desired neck angle, $N_{\theta,max}$ and $N_{\theta,min}$ are maximum and minimum of neck angle, and $T_{h,max}$ and $T_{h,min}$ are maximum and minimum of table heights. The T_h is the current table height described in Fig. 3. In the actual environment, if a robot detects a very low table height (less than 70 cm), it positions itself very close to the table. As a result, the workspace for manipulation reduces and difficult to manipulate. To establish an appropriate area for grasping an object, it secures the workspace using a laser sensor to measure the distance between the table and robot base. After the robot judges that the workspace is enough to manipulate then detection and segmentation of the multiple objects start. This process is represented in the flow chart (See the Fig. 4).

where $N_{\theta,d}$ is the desired neck angle, $N_{\theta,max}$ and $N_{\theta,min}$ are maximum and minimum neck angle, respectively, and $T_{h,max}$ and $T_{h,min}$ are maximum and minimum table heights, respectively. The T_h is the current table height described in Fig. 3. In the actual environment, if a robot detects a very low table height (less than 70 cm), it positions itself very close to the table. As a result, the workspace for manipulation is reduced, which makes it difficult to manipulate. To establish an appropriate area for grasping an object, the robot secures the workspace using a laser sensor to measure the distance between the table and robot base. After the robot judges that the workspace is appropriate for manipulation, detection and segmentation of the multiple objects starts. This process is represented in the flow chart in Fig. 4).

5.1. Multi-object segmentation

When the robot explores its environment and encounters multiple objects, a depth-based segmentation algorithm could be useful for extracting the objects using point cloud data acquired from the RGB camera which is first implemented by [24–26]. Trevor *et al.* proposed a connected component-based approach to segment a set of objects from an organized point cloud [24].

In our work, we adapted and modified the Trevor *et al.*'s approach to segment multiple objects by an organized point cloud library. We suggest to use the depth camera (Xtion) for acquiring depth data instead of RGB camera will increase depth data accuracy. For each point $P(x,y)$ a label $L(x, y)$ is assigned. Points belonging to the same segment will be assigned to the same label based on Euclidean clustering comparison function (See the [24] for more details). To segment the objects accurately, some of the large segments, like the plane surface, will be excluded. In addition, if the distance between the two points in the same label set is more than a threshold, one of the points will be discarded because of increasing object segmentation speed. The area to be clustered and the threshold of points for each object between 1500 to 10000 points were chosen experimentally. In order to distinguish multiple objects easily, the object were covered with six RGB colors. The result of this segmentation process is presented in Fig. 5. This process is described in the flow chart (See the Fig. 4).

In our work, we adapted and modified the approach of *et al.* to segment multiple objects by an organized point cloud library. We used a depth camera (Xtion) for acquiring depth data instead of an RGB camera to increase depth data accuracy. For each point $P(x,y)$, a label $L(x, y)$ is assigned. Points belonging to the same segment will be assigned to the same label based on the Euclidean clustering comparison function (see [24] for more details). To segment the objects accurately, some of the large segments, like the plane surface, will be excluded. In addition, if the distance between the two points in the same label set is more than a threshold, one of the points will be discarded because of increasing object segmentation speed. The area to be clustered and the threshold of points for each object. The points of each object clustered between 1500 to 10000 points were chosen experimentally. To distinguish between multiple objects easily, the object were covered with six RGB colors. The result of this segmentation process is presented in Fig. 5. This process is described in the flow chart shown in Fig. 4).

5.2. Human object selection

Autonomous object selection is the best method for humans. However, selection of the target object in the domestic environment is a difficult task, although multiple objects were clustered completely by a camera. For this reason, the selection was conducted via human intelligence. A variety of interfaces for object selection have been developed [12–16]. For our robotic platform, object selection was done in two ways: 1) voice and 2) visual. We also believe that a combination of these two methods could be easily accessible for very old people who cannot move. Moreover, our interface platform includes a tablet for the voice user interface (see Fig. 6(b)) and rviz in a PC for the visualization interface (see Fig. 6(a)). The visualization interface, which includes RGB colors and depth information of the environment, was provided for the selection system; the voice user interface is based on speech recognition [28]. The selection system consisted of three steps:

- 1. Select one of the rows of multiple objects
- 2. Choose an object desired in the same row
- 3. Choose an object desired in a different row

First, the user selects the object from the front or back row of the table. After identification of the row (front or back), the target object selection is done (see Fig. 4).

5.3. Action planning & Execution

Among multiple objects, grasping objects are still a challenge. Thus, we tried to find a grasping point with simply shaped objects such as a bottle or box, which are common household objects in

Figure 5. Real-time multi-object segmentation on the table visualized in rviz [27]

the domestic environment. In addition, the grasping point was used to generate possible hand poses relative to the object for action planning.

To extract the grasping point from each object, we used the 3D centroid function in PCL and configured the grasp poses. In our case, we characterized two types of grasp poses:

- Top pose: It is aligned by the robot hand to the object in the vertical plane (along the x- and z-axis), and opening and closing of the robotic hand is in the direction of the x- or y-axis (see Fig. 7(a)).
- Side pose: It is defined in the horizontal plane (along the x- and y-axis), and the opening and closing direction of the robotic hand is the same as previous (see Fig. 7(b)).

To grasp the object, we used the motion planning library, which includes capability for collision avoidance, self-collisions, and joint limit avoidance of the robot arm in the domestic environment. The motion planning library (Moveit) was used for executing three mobile manipulation strategies. In addition, the library supports collision-aware inverse kinematics to determine the feasibility of grasp by finding collision-free solutions. Even if collision-free solutions were found, many possible paths could be used to reach the goal, and the library choose a path generated with minimum trajectory cost; namely, the shortest in the joint space. In addition, the position of the robot hand plays an important role in grasping. For this reason, pre-grasp position (it is an offset from the target object with the two grasp poses) was developed. After the pre-grasp position was obtained, the palm of the robot hand approached the surface of the target object to grasp it. Based on these technologies, the strategies were enhanced to avoid crashes between the robot arm and robot body during the operation [29] (see Fig. 4).

Figure 6. (a) Visualization interface system, (b) Voice user interface system

Figure 7. (a) Top grasp pose, (b) Side grasp pose

5.4. Developed mobile manipulation strategies

Three strategies of the mobile manipulation were conceived to grasp an object, which was apart from the robot. A set of 6 objects, arranged in two rows, was placed in front of the robot (see Fig. 6(a)). We consider grasping objects placed in the back row because grasping front row objects are an easy task that we have already developed. Before starting the strategies for grasping an object, we need to accomplish three steps. The first step is initialization of the robot arm. The next step is to transform the coordinates of multiple objects from camera frame to robot base frame for manipulation. The last step is pre-grasp position based on table height. These three steps are described in Algorithm 1 (lines 2 to 5). Actually, these steps are capable of grasping an object on a table, but grasping back-row objects always fails due to the obstruction caused the front-row objects. For these reasons, we developed three strategies for the mobile manipulation for grasping an object in the back row. To exploit the three strategies of mobile manipulation, motion planning with Moveit library is applied to prevent the robot arm from colliding with the robot body.

• The first strategy

The objective of the **first strategy** was to grasp an object on the approximately 70cm high table, directly from the back row, to reduce manipulation time. The mobile platform was pre-defined to be at a rotated angle and also the specific neck angle that supports segmentation of objects in the back

Figure 8. The first strategy for manipulation: (a) The robot moves close to the table, (b) The mobile platform is rotated to grasp the target object. (c) Top grasp pose is implemented to grasp the object directly. (d) After grasping the object, the robot arm returns back to the initial position.

row was set. In fact, the two angles were defined empirically based on distance between the table and robot. However, the information of the scene obtained by the camera was still insufficient due to the obstruction of front row objects. Actually, when the same sizes of objects are detected, the visualization of the object size shows differently because the distance from the camera to each object is different. In addition, the objects in the front row would be obstructed during grasping, and it will be difficult to detect the entire size of the back row objects. For this reason, the function of the linear interpolation (the same as Equation 1) with different variables was developed. The output of the interpolation is a value that will add to the position in the z-axis to establish the stable grasping point. After the interpolation, the top grasp pose was applied to grasp the object directly. The first strategy in the actual environment was performed as shown in Fig. 8((a)-(d)) and also as described in Algorithm 1 (lines 7 to 9).

• **The second strategy**

The first strategy of the mobile manipulation was useful to grasp the objects in the back. However, we still are challenged to ensure stable grasping by the robot. For this reason, we developed a new strategy to grasp the objects in the back row to compensate for an inadequate object segmentation and robust and stable grasping. The objective of the **second strategy** was to grasp the objects from an 80-cm-high table while ensuring good stability. To pick up the objects, the algorithm for removing the objects in the front was conceived. The point of the strategy is that when the user selects the back row and target object, the robot calculates the centroid front row object as well. Then, the robot lifts the object off the front row and places it in the empty place on the table. First, to find the objects in the front, the function was implemented for searching the nearest distance between all objects and the target object. After the object in the front row was found, the pre-defined grasp position was applied.

Algorithm 1: The three mobile manipulation strategies

Input : Joint position $\mathbf{q}$, Initial pose $\mathbf{x}_{init}$, Object row O_{row} , All objects information O_{all} , The centroid of the object selected and transformed O_{cen} , Object desired $\mathbf{x}_d$, Grasp pose $\mathbf{x}_{grasp}$, The new centroid of the object selected and transformed O_{newcen} , Distance of z axis $\mathbf{z}_{add}$, Table height T_h

Output: Goal pose, $\mathbf{x}_{goal}$

```

1 while until manipulator finish task given do
2   InitializationJacoArm( $\mathbf{q}$ );
3   TransformAllObjects;
4    $\mathbf{x}_{grasp} \leftarrow$  Pre-definedGraspPose( $\mathbf{x}_{init}$ ,  $O_{row}$ ,  $T_h$ );
5   if  $O_{row}.back = True$  then
6     1. The first strategy
7     Rotate MobileBase & Set the Angle of Neck
8      $\mathbf{z}_{add} \leftarrow$  InterpolationAddGraspAxis( $O_{cen}$ );
9      $\mathbf{x}_{goal} \leftarrow \mathbf{x}_{cen} \cdot \mathbf{x}_{grasp} \cdot \mathbf{z}_{add}$ ;
10    2. The second strategy
11    Search NearestObject( $O_{cen}$ );
12    Search Pre-defiendEmptySpace( $O_{all}$ );
13     $\mathbf{x}_d \leftarrow$  Calculate TargetObject( $O_{cen}$ );
14     $\mathbf{x}_{goal} \leftarrow \mathbf{x}_d \cdot \mathbf{x}_{grasp}$ ;
15    3. The third strategy
16    Update ObjectStates & Move MobileBase;
17     $O_{newcen} \leftarrow$  Search ObjectUsingAxis( $O_{cen}$ );
18     $\mathbf{x}_d \leftarrow$  Calculate TargetObject( $O_{newcen}$ );
19    ManipulationBasedOnModeSelection;
20     $\mathbf{x}_{goal} \leftarrow \mathbf{x}_d \cdot \mathbf{x}_{grasp}$ ;
21  end
22 end

```

Figure 9. The second strategy shows the robot performing lateral grasp to pick up an object in the back. (a) First, the object in front of the object selected is removed. (b) The object is placed on the empty place. (c) The robot grasps the target object with a side grasp pose. (d) As the object grasps, the arm starts to return back to the initial position.

To ensure a stable grasping, the side grasp pose was introduced. Then, as the front row object was grasped (see Fig. 9(a)), a pre-defined place was located at the right edge of the table (see Fig. 9(b)). Since the robot arm is mounted on the right of the body, we considered that an available place to the right would be easier. After the object was placed on the table, the arm returned to the initial position and the robot started grasping the target object with the side grasp pose (see Fig. 9(c,d)). The entire process is represented in Algorithm 1 (lines 11 to 14).

• The third strategy

The first and second strategies of the mobile manipulation were helpful to grasp objects in the back row, but the robot might fail to accomplish the task. For example, if the robot faced a table higher than its visual field, or if objects in the front row were taller than objects in the back row, the robot could not detect objects in the back row. For this reason, the **third strategy** was developed to grasp the hidden object. The third strategy is used in the particular situation of a hidden object when the first and second strategies cannot perform grasping tasks. In this strategy, human support was exploited to overcome the difficulty of object selection. For instance, a human can evaluate placement of the object better than a robot can. In other words, if the user cannot see the object by visual interface, they can suggest other alternatives. To conduct a feasibility study for the third strategy, the hidden object was evaluated according to the decision of the user. In the first and second strategies, the user selects an object in the back row and the target object. The object selection method in the third strategy was not the same as in previous strategies because the user cannot see the object in the back using visual interface. However, the user already knows the location of the target object on the table and selects the back row and an object in the front using voice interface. After the user selects both row and object, the

Figure 10. Experimental setup with multiple objects and an adjustable height table.

process of the third strategy, which is similar to that of the second strategy, is implemented. The basic difference between these two strategies is to update the state of the objects. After the object in the front is placed in the empty space, the robot should discover the target object. To find the object, the state of the scene was updated using the multi-object segmentation function. In addition, the information of the y-axis is used to find the target object because the target object is located colinear to the front object. The simplified algorithm is described in Algorithm 1 (lines 16 to 20).

6. Experimental setup

Our experimental setup is shown in Fig. 10. The robotic platform for the experiment is the Doro (domestic robot) personal robot [7], a service robot equipped with an omni-directional mobile base and a Kinova Jaco robotic arm. The Kinova Jaco arm, which has six degrees of freedom, is used for manipulation tasks. The head of the Doro is a pan-tilt platform equipped with two stereo cameras and an Asus Xtion Pro depth camera; they are used for object detection and segmentation. Furthermore, front laser (SICK S 300) and laser (Hokuyo URG-04LX) sensors are used to detect obstacles for safe navigation in a domestic environment. The three manipulation strategies were tested in the DomoCasa Lab, a domestic house developed and managed by The BioRobotics Institute of Scuola Superiore Sant'Anna in Peccioli, Italy. To implement ADL, we set up the experimental environment with multiple objects placed on the table, which can be adjusted in height as shown in Fig. 10. Three objects were

Figure 11. Objects were placed as follows: (a) left top: short objects, right: tall objects; (b) short objects in the front (FSO, right top); (c) tall objects in the front (FTO, left bottom); and (d) random size objects (RO, right bottom).

placed in the front, and the others were placed in the back. We used rectangular objects such as plastic bottles and juice boxes during the experiments.

For the experiment, several scenarios were organized. Before grasping an object in the back row, we tested a simple scenario for grasping an object in the front row. Then, the three manipulation strategies were tested to grasp an object in the back. The known table height was set at different steps of 10 cm, such as 70, 80, 90, and 100 cm. In addition, these three strategies were tested at an unknown table height to apply them in the real-life situation. The objects were placed in three positions: short size objects in the front (FSO), tall size objects in the front (FTO), and random size objects (RO) (see Fig. 11). During the manipulation with all strategies for unknown table height, we considered grasping one of the objects, which were placed randomly. The scenarios were evaluated 10 times for each strategy in terms of collision, execution time, and success rate with known table height and with unknown table height.

7. Experimental results

Firstly, quantitative and qualitative analysis for three mobile manipulation strategies for known and unknown table height were performed. We only considered the execution time and collisions when the mobile manipulation task was a success.

7.1. Quantitative analysis at known table heights

The quantitative results focused on three criteria: success rates, collision, and execution time. The success rates were measured when the robot grasped a target object. We also considered a collision case in which objects were crashed into by a robot hand. The execution time is measured beginning when these experiments start.

7.1.1. Success rates

The success rates were evaluated in each strategy with three different object positions (FSO, FTO, and RO) on known table heights. When the kinematic solver operating in Moveit could not work properly, we did not measure the success rate during trials.

As shown in Fig. 12(a), the success rate of the first strategy was higher for the 70 cm table height compared to any other table heights and strategies (60% to 80% for all three scenarios). At this table

height, the second and third strategies have low success rates of manipulation because of lack of workspace. However, if the robot faces the same situation, the first strategy can grasp objects with top grasp pose, which helps the robot to avoid front obstacles and save manipulation time. Nevertheless, some trials in the first strategy also failed to grasp the target object, although we developed linear interpolation to overcome insufficient object segmentation and stable grasping. In addition, except for the 70 cm table height, the first strategy was not successful in grasping because the robot cannot reach pre-grasp position over 77 cm.

The success rate of the second strategy was improved for a table height of greater than 80 cm, which is better than for the first strategy, but it doesn't work for the 70 cm table height (see Fig. 12(b)). In particular, we found that the second strategy was a success for an average of 30% of the 70 cm table height tests in three different scenarios. Moreover, this strategy performed better for 80, 90, and 100 cm table heights for FSO and RO (success rate varies from 60% to 8%). We also observed that the second strategy failed to grasp FTO objects at table heights of 90 and 100 cm, which occurred due to taller objects blocking the target object (the human could not see or select the target object). In addition, when the robot grasped an object in the FTO, the robot only segmented small parts of objects in the back at 80 cm table height. Therefore, the grasp point was not extracted accurately.

Finally, the third strategy (see Fig. 12(c)) could be carried out with any table height. In this case, the success rate varies from 70% to 80% (higher than second strategy) at 80, 90, and 100 cm table heights. However, for the 70 cm table height, the performance is similar to that of the second strategy (20% to 30%). As the robot removed the front object, the multi-object segmentation system was repeated automatically. As a result, the grasp point could be extracted more accurately than with the second strategy. However, as mentioned before, we still have a kinematic solver problem. Therefore, when the kinematic solver performed well, we deemed the manipulation to be successful. Failure of the strategy occurred when the grasp force was insufficient to grasp the target object. Thus, the robot dropped the object during manipulation. Furthermore, if the grasp point of the object was slightly shifted from the centroid of the object, the robot arm could not grasp the object. As shown in Fig. 12(c), the third strategy can be applied in any environment and shows better performance except for the 70 cm table height.

7.1.2. Collisions

During the evaluation, the number of collisions was measured for each table height using the three strategies for a total of 10 times for all scenarios in the experiments (see Fig. 13).

The best results with 70 cm table height were achieved using the first strategy with a total average of seven collisions from all scenarios (see Fig. 13(a)). The collisions in the strategy occurred while the robot arm returned to the home position. Except for the 70 cm table height, the low number of collisions occurred at 80, 90, and 100 cm table height with the third strategy. The total number of collisions using the strategy occurred with all scenarios, with averages of six, eight, and ten for 80, 90, and 100 cm table height respectively (see Fig. 13(b),(c),(d)), and standard deviation is about 5% of each collision. However, the second and third strategies have similar manipulations. Therefore, Fig. 13(b),(c),(d) show that the collisions of the strategies are similar except for 90 cm and 100 cm table heights in the FTO scenario. The collisions with two strategies occurred while the robot arm was close to the object and returned to the home position with a target object.

Actually, with the first strategy, collisions only with the 70 cm table height could be measured because the robot arm could not reach objects with the other table heights (see Fig. 13(a)). Moreover, we could not measure collisions with the second strategy in the FTO at the 90 and 100 cm table heights since the objects in the back were occluded due to being shorter than the front objects (see Fig. 13(c),(d)).

Figure 12. The success rate of the mobile manipulation for grasping an object in the back row on known table heights: (a) First strategy; (b) Second strategy; (c) Third strategy.

7.1.3. Execution times

We measured the execution times for grasping objects in the back row using three strategies. Then we calculated the average time with FSO, FTO, and RO configurations. The three categories for the time measured were separated (see Fig. 14):

- 1. Image preprocessing and multi-object segmentation (IP & MOS)
- 2. Human object selection (HOS)
- 3. Each strategy execution time (ESE)

In three strategies, the execution time for the first and the second categories were similar. The first category took 35 seconds to finish, but the robot can finish the task in less than 25 seconds. Actually, sometimes the depth camera did not detect objects because of noise from input data. Therefore, the robot repeats segmentation with new input data, which takes more time (~ 10 seconds).

In addition, when a user selects a row and an object, it spends less than 10 seconds. However, sometimes the voice interface could not be recognized by the tablet directly. Thus, we used time to request the object row and target object again (~ 15 seconds).

Comparing each strategy execution time (third category), the first strategy is the fastest because the strategy does not consider removing the front object. In contrast, the third strategy is slower than the second strategy since the third strategy needs more time to segment multiple objects due to repetition.

Figure 13. Number of collisions with three strategies during the grasp of an object in the back row for known table heights: (a) 70 cm table height; (b) 80 cm table height; (c) 90 cm table height; (d) 100 cm table height.

7.2. Quantitative results in unknown table heights

Previous quantitative results were analyzed using known table height. However, various types of tables exist in reality. Before we set up the table height, we defined range between 70 and 100 cm to select strategies automatically. Then, the table height was set up randomly between defined ranges. Also, we only tested the strategies with objects in the RO configuration for implementing in the actual environment.

To confirm the three strategies of mobile manipulation, the three different table heights were measured and the results were evaluated in the same manner as previous cases (see Fig. 15). The robot selected one strategy automatically to manipulate according to table height. As a result, the experiment was tested in ten trials; the average of the success rate of the manipulation in unknown table height is greater than 75%. We analyzed the number of collisions during the experiment. Collisions were evaluated with the same criteria, and an average of five collisions occurred with all three scenarios. In addition, the execution time for self-selected strategies at unknown table height (73.3 cm, 84.3 cm, and 93.7 cm) are 104 s, 178 s, and 184 s for the random scenario (see Fig. 16). Also, this result is only measured when the kinematic solver operates well.

7.3. Qualitative results for known and unknown table heights

For known table height, the first strategy was developed in terms of saving time. However, in terms of stability, the second and third strategies performed better than the first strategy. In addition, the voice and the visual interfaces together made it more comfortable and convenient for the user to select a target object. Specifically, sometimes a few users are not able to see the object, and in that case they can ask the robot vocally (more likely with elderly people and children).

Figure 14. Visualization of execution times for grasping an object on known table heights with three strategies.

Figure 15. The success rate and number of collisions with three strategies of the mobile manipulation for grasping an object in the back row with unknown table heights.

For unknown table height, the robot demonstrated that self-selection strategies succeeded by enabling less user input and more intelligent selection. Moreover, unknown table height with random object selection is a better fit for the domestic environment. Upon receiving better performance by the robot with intelligent selection, the user will be more comfortable with the robot and find it more convenient.

8. Conclusions and future work

In this paper, we present three mobile manipulation strategies in which the operator provides a simple command using visual and voice user interfaces. The strategies resulted in the improvement of grasping capabilities of household objects in collaboration with a real robot Doro. The user provides the two commands regarding the object row and target object using a visual and a voice interface. The three strategies of the mobile manipulation were developed to pick and place, and convey an object in the domestic environment effectively. Based on the results, the three strategies have their own advantages at the different table heights. Therefore, the intelligent strategy selection system can be applied for domestic environments that have different table heights.

The goal of this paper is to support elderly people for ADL in the domestic environment. Although the proposed system considered grasping limited to certain types of domestic objects, the strategies we developed can apply grasping to various household objects. In addition, to take care of elderly people daily, monitoring and managing systems using robots are invaluable. In this sense, our proposed

Figure 16. Visualization of execution times for grasping an object with unknown table heights using three strategies.

system can be useful to monitor the robot state and select an object easily for ADL. Nevertheless, we still have many issues, including detection and segmentation, that we need to resolve in the domestic environment. Actually, the current system could be used to detect, cluster, and extract simple household objects such as bottles, boxes, etc. However, various objects that are different in shape exist in the domestic environment. Therefore, the 3D centroid of an object would not be able to grasp it. For this reason, we will develop a grasp pose algorithm for a variety of household objects with our strategies to save time [30]. In addition, a deep learning-based approach for extracting grasping point could be considered to obtain more accurate performance [31,32]. Moreover, for the insufficient shape information, the tactile sensors could be used on the fingers of the robotic hand. These would help the Doro robot for contact-reactive grasping of objects [33].

Acknowledgments: The work described was supported by the Robot-Era and ACCRA project, respectively founded by the European Community’s Seventh Framework Programme FP7-ICT-2011 under grant agreement No. 288899 and the Horizon 2020 Programme H2020-SCI-PM14-2016 under grant agreement No. 738251.

References

1. Aquilano, M.; Cavallo, F.; Bonaccorsi, M.; Esposito, R.; Rovini, E.; Filippi, M.; Esposito, D.; Dario, P.; Carrozza, M.C. Ambient assisted living and ageing: Preliminary results of RITA project. Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE. IEEE, 2012, pp. 5823–5826.
2. Cavallo, F.; Aquilano, M.; Bonaccorsi, M.; Mannari, I.; Carrozza, M.; Dario, P. Multidisciplinary approach for developing a new robotic system for domiciliary assistance to elderly people. Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE. IEEE, 2011, pp. 5327–5330.
3. Turchetti, G.; Micera, S.; Cavallo, F.; Odetti, L.; Dario, P. Technology and innovative services. *IEEE pulse* **2011**, 2, 27–35.
4. Schraft, R.; Schaeffer, C.; May, T. Care-O-bot/sup TM: the concept of a system for assisting elderly or disabled persons in home environments. Industrial Electronics Society, 1998. IECON’98. Proceedings of the 24th Annual Conference of the IEEE. IEEE, 1998, Vol. 4, pp. 2476–2481.
5. Cavallo, F.; Limosani, R.; Manzi, A.; Bonaccorsi, M.; Esposito, R.; Di Rocco, M.; Pecora, F.; Teti, G.; Saffiotti, A.; Dario, P. Development of a socially believable multi-robot solution from town to home. *Cognitive Computation* **2014**, 6, 954–967.
6. Torta, E.; Oberzaucher, J.; Werner, F.; Cuijpers, R.H.; Juola, J.F. Attitudes towards socially assistive robots in intelligent homes: results from laboratory studies and field trials. *Journal of Human-Robot Interaction* **2012**, 1, 76–99.
7. Dogar, M.; Srinivasa, S. A framework for push-grasping in clutter. *Robotics: Science and systems VII* **2011**, 1.

8. Kitaev, N.; Mordatch, I.; Patil, S.; Abbeel, P. Physics-based trajectory optimization for grasping in cluttered environments. *Robotics and Automation (ICRA)*, 2015 IEEE International Conference on. IEEE, 2015, pp. 3102–3109.
9. Stilman, M.; Schamburek, J.U.; Kuffner, J.; Asfour, T. Manipulation planning among movable obstacles. *Robotics and Automation*, 2007 IEEE International Conference on. IEEE, 2007, pp. 3327–3332.
10. Ozgür, A.; Akin, H.L. Planning for Tabletop Clutter Using Affordable 5-DOF Manipulator.
11. Fromm, T.; Birk, A. Physics-based damage-aware manipulation strategy planning using Scene Dynamics Anticipation. *Intelligent Robots and Systems (IROS)*, 2016 IEEE/RSJ International Conference on. IEEE, 2016, pp. 915–922.
12. Pitzer, B.; Styer, M.; Bersch, C.; DuHadway, C.; Becker, J. Towards perceptual shared autonomy for robotic mobile manipulation. *Robotics and Automation (ICRA)*, 2011 IEEE International Conference on. IEEE, 2011, pp. 6245–6251.
13. Ciocarlie, M.; Hsiao, K.; Leeper, A.; Gossow, D. Mobile manipulation through an assistive home robot. *Intelligent Robots and Systems (IROS)*, 2012 IEEE/RSJ International Conference on. IEEE, 2012, pp. 5313–5320.
14. Leeper, A.E.; Hsiao, K.; Ciocarlie, M.; Takayama, L.; Gossow, D. Strategies for human-in-the-loop robotic grasping. *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 1–8.
15. Witzig, T.; Zöllner, J.M.; Pangercic, D.; Osentoski, S.; Jäkel, R.; Dillmann, R. Context aware shared autonomy for robotic manipulation tasks. *Intelligent Robots and Systems (IROS)*, 2013 IEEE/RSJ International Conference on. IEEE, 2013, pp. 5686–5693.
16. Birkenkamp, P.; Leidner, D.; Borst, C. A knowledge-driven shared autonomy human-robot interface for tablet computers. *Humanoid Robots (Humanoids)*, 2014 14th IEEE-RAS International Conference on. IEEE, 2014, pp. 152–159.
17. Gualtieri, M.; Kuczynski, J.; Shultz, A.M.; Ten Pas, A.; Platt, R.; Yanco, H. Open world assistive grasping using laser selection. *Robotics and Automation (ICRA)*, 2017 IEEE International Conference on. IEEE, 2017, pp. 4052–4057.
18. Laskey, M.; Lee, J.; Chuck, C.; Gealy, D.; Hsieh, W.; Pokorný, F.T.; Dragan, A.D.; Goldberg, K. Robot grasping in clutter: Using a hierarchy of supervisors for learning from demonstrations. *Automation Science and Engineering (CASE)*, 2016 IEEE International Conference on. IEEE, 2016, pp. 827–834.
19. Parasuraman, R.; Sheridan, T.B.; Wickens, C.D. A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans* **2000**, 30, 286–297.
20. Davies, E.R. *Machine vision: theory, algorithms, practicalities*; Elsevier, 2004.
21. Rusu, R.B.; Cousins, S. 3d is here: Point cloud library (pcl). *Robotics and automation (ICRA)*, 2011 IEEE International Conference on. IEEE, 2011, pp. 1–4.
22. Chitta, S.; Sucan, I.; Cousins, S. MoveIt! [ROS topics]. *IEEE Robotics & Automation Magazine* **2012**, 19, 18–19.
23. Fischler, M.A.; Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **1981**, 24, 381–395.
24. Trevor, A.J.; Gedikli, S.; Rusu, R.B.; Christensen, H.I. Efficient organized point cloud segmentation with connected components. *Semantic Perception Mapping and Exploration (SPME)* **2013**.
25. Ersen, M.; Ozturk, M.D.; Biberici, M.; Sariel, S.; Yalcin, H. Scene interpretation for lifelong robot learning. *Proceedings of the 9th international workshop on cognitive robotics (CogRob 2014) held in conjunction with ECAI-2014*, 2014.
26. Beksi, W.J.; Papanikolopoulos, N. 3D region segmentation using topological persistence. *Intelligent Robots and Systems (IROS)*, 2016 IEEE/RSJ International Conference on. IEEE, 2016, pp. 1079–1084.
27. Hershberger, D.; Gossow, D.; Faust, J. RViz, 2011.
28. Sak, H.; Senior, A.; Rao, K.; Beaufays, F.; Schalkwyk, J. Google voice search: faster and more accurate. *Google Research blog* **2015**.
29. Ciocarlie, M.; Hsiao, K.; Jones, E.G.; Chitta, S.; Rusu, R.B.; Şucan, I.A. Towards reliable grasping and manipulation in household environments. *Experimental Robotics*. Springer, 2014, pp. 241–252.
30. Redmon, J.; Angelova, A. Real-time grasp detection using convolutional neural networks. *Robotics and Automation (ICRA)*, 2015 IEEE International Conference on. IEEE, 2015, pp. 1316–1322.

- 558 31. Lenz, I.; Lee, H.; Saxena, A. Deep learning for detecting robotic grasps. *The International Journal of Robotics*
559 *Research* **2015**, *34*, 705–724.
- 560 32. Levine, S.; Finn, C.; Darrell, T.; Abbeel, P. End-to-end training of deep visuomotor policies. *Journal of*
561 *Machine Learning Research* **2016**, *17*, 1–40.
- 562 33. Hsiao, K.; Chitta, S.; Ciocarlie, M.; Jones, E.G. Contact-reactive grasping of objects with partial shape
563 information. *Intelligent Robots and Systems (IROS)*, 2010 IEEE/RSJ International Conference on. IEEE,
564 2010, pp. 1228–1235.