EL-E: An Assistive Mobile Manipulator that Autonomously Fetches Objects from Flat Surfaces

Advait Jain · Charles C. Kemp

Received: 2 February 2009 / Accepted: 19 August 2009

Abstract Assistive mobile robots that autonomously manipulate objects within everyday settings have the potential to improve the lives of the elderly, injured, and disabled. Within this paper, we present the most recent version of the assistive mobile manipulator EL-E with a focus on the subsystem that enables the robot to retrieve objects from and deliver objects to flat surfaces. Once provided with a 3D location via brief illumination with a laser pointer, the robot autonomously approaches the location and then either grasps the nearest object or places an object. We describe our implementation in detail, while highlighting design principles and themes, including the use of specialized behaviors, task-relevant features, and low-dimensional representations.

We also present evaluations of EL-E's performance relative to common forms of variation. We tested EL-E's ability to approach and grasp objects from the 25 object categories that were ranked most important for robotic retrieval by motor-impaired patients from the Emory ALS Center. Although reliability varied, EL-E succeeded at least once with objects from 21 out of 25 of these categories. EL-E also approached and grasped a cordless telephone on 12 different surfaces including floors, tables, and counter tops with 100% success. The same test using a vitamin pill (ca. 15mm ×5mm ×5mm) resulted in 58% success.

Keywords Mobile Manipulation · Assistive Robotics

1 Introduction

For millions of people on a daily basis, motor impairments diminish quality of life, reduce independence, and increase healthcare costs. Assistive mobile robots that autonomously manipulate objects within everyday settings have the potential to improve the lives of the elderly, injured, and disabled by augmenting their abilities with those of a cooperative robot.

Advait Jain · Charles C. Kemp
Georgia Institute of Technology, Atlanta, GA, USA
E-mail: advait@cc.gatech.edu, charlie.kemp@bme.gatech.edu

This pdf is an author created version. The original publication is available from www.springerlink.com (DOI 10.1007/s10514-009-9148-5)

Fig. 1: The mobile manipulator, EL-E, delivering an object to a table after the patient briefly illuminated it with a laser pointer (image used with permission and IRB approval).

Within this paper we present the most recent version of EL-E (pronounced "Ellie"), an assistive mobile manipulator that we are actively developing to help people with motor impairments in everyday tasks (Figure 1). People with motor impairments have consistently placed a high priority on retrieving objects from the floor and shelves (Stanger et al, 1994), so a robot capable of performing pick and place operations within human environments would be valuable. Moreover, it could serve as a foundation upon which other forms of assistance could be built.

For this paper, we focus on the subsystem that enables EL-E to grasp an object from a flat surface, carry it to a new location, and then place it on a flat surface. The user designates the location at which to grasp an object or place a previously grasped object by briefly illuminating the location with a laser pointer. Since flat planes that are orthogonal to gravity are common in indoor human environments and many objects are found on these planes, we expect for this capability to be useful.

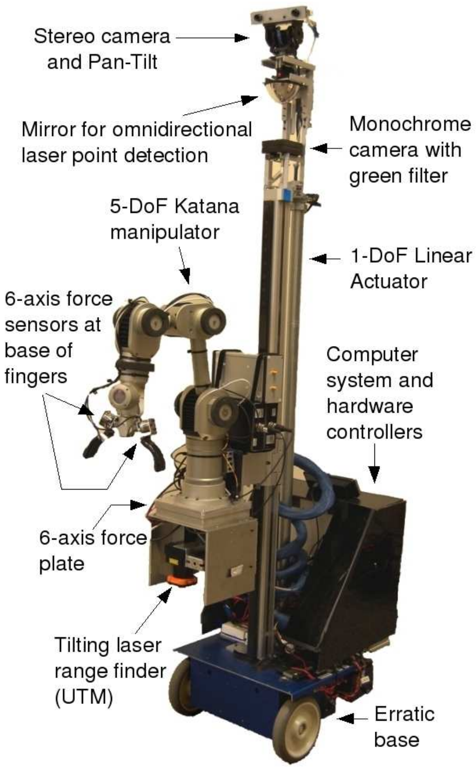


Fig. 2: The mobile manipulator, EL-E, used in this paper.

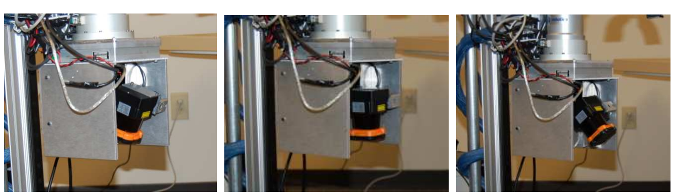


Fig. 3: This figure shows the servo tilting the laser range finder to generate a 3D point cloud.

We describe this subsystem in detail, while highlighting more general design principles and themes. For example, we show how EL-E exploits the structure of indoor human environments with its body and behaviors. We also emphasize the potential benefits of using specialized behaviors and task-relevant features. EL-E performs navigation and grasping using behaviors that are modular, composable, predictable, and robust. These behaviors use low-dimensional features such as a 3D location near the object to be grasped, the edge of the surface that supports the object, and a 3 dimensional feature vector that represents the object for grasping. Similarly, EL-E uses a low-dimensional action representation that parameterizes overhead grasps in terms of the gripper's 2D planar position and 1D wrist orientation. Rather than using global models of the environment, EL-E focuses its perceptual resources on a spatially local volume of interest (VOI). This attention system contributes to EL-E's ability to operate in diverse environments by ignoring non-local information. Instead of using pre-existing models of specific objects, EL-E segments and grasps objects without explicit models.

We also present evaluations of EL-E's performance relative to common forms of variation, including object geometry, color, material properties, and surface characteristics. For example, Figure 4 shows EL-E successfully approaching and grasping a vitamin pill in the unmodified break room at the Health Systems Institute at Georgia Tech. In addition to presenting the raw results of our tests, we analyze the common forms of failure to gain insight into how the system could be improved.

2 Related Work

Within this section, we give a brief overview of related work along with comparisons to our system.

2.1 Prior Research with EL-E

We have previously presented results related to EL-E's user interface and method of behavior selection (Kemp et al, 2008; Nguyen et al, 2008b). We have also reported results on a user study in which patients from the Emory ALS[1] Center successfully designated objects for EL-E to pick up (Choi et al, 2008). Additionally, we have published a brief overview of the original version of EL-E, which included a small number of tests (7 trials total) (Nguyen et al, 2008a). Since then, EL-E's performance has greatly improved and EL-E has changed substantially both in hardware and software. For example, EL-E now uses a tilting laser range finder (see Figure 3) and no longer has an eye-in-hand camera.

2.2 Robots for Assistive Manipulation

There is a long history of research into assistive robots for people with motor impairments (Van der Loos and Reinkensmeyer, 2008). Researchers have developed stationary workstations for assistance in offices and factories (Dallaway and Jackson, 1992; Van der Loos, 1995; Van der Loos et al, 1999), and eating and drinking (Topping and Smith, 1998). Wheelchair mounted robot arms have provided mobile assistance and are beginning to exhibit more autonomy (Kwee et al, 1989; Hernandez et al, 2008; Tsui et al, 2008).

[1] Amyotrophic lateral sclerosis (ALS), sometimes referred to as Lou Gehrig's Disease, is a progressive neurodegenerative disease that causes people to gradually lose the ability to move.

Fig. 4: These images are from one of the experiments that we performed to evaluate the performance of the robot (Section 5.1). The robot picks up a pill from a kitchen counter top after the user briefly illuminates the pill with a laser pointer. The first image shows the starting location of the robot. The robot approaches the pill in a direction normal to the surface, segments the pill and positions the gripper above the pill. The gripper then moves down until it makes contact with the surface and picks up the pill.

Mobile manipulators have been a relatively rare form of assistive robot. The MoVAR and MOVAID projects represent early work in this area (Van der Loos, 1995; Dario, 1999). Recent research has sought to develop assistive mobile manipulators with autonomy using methods such as model-based planning and visual servoing (Graf et al, 2004; Remazeilles et al, 2008).

Unlike other approaches to robotic assistance, an autonomous mobile manipulator could perform tasks independently from the patient, would not require donning and doffing, and would not directly encumber the patient. It could also assist a wide variety of patients (e.g., a patient does not need to be missing a limb or using a wheelchair to benefit).

2.3 Autonomous Mobile Manipulation

There has been a surge of interest in autonomous mobile manipulation in human environments (Grupen and Brock, 2004; Kemp et al, 2007; Khatib et al, 1999; Brock and Khatib, 2002; Zllner et al, 2004; Bluethmann et al, 2004; Waarsing et al, 2001; Inamura et al, 2008; Okada et al, 2005; Hillenbrand et al, 2004; Kragić et al, 2002).

Most research systems have made use of pre-defined models of the environment and objects, which the robot registers to its sensory data, such as in Srinivasa et al (2008) and Asfour et al (2007).

In contrast to work that uses detailed geometric models of the environment with planning, EL-E uses sparse, low-dimensional, task-relevant features with specialized behaviors. EL-E uses its sensors to perceive these features when they are needed. Consequently, EL-E can immediately begin operating in a new environment with new objects, since neither a map of the environment nor complete geometric models of the objects are required. This approach is in the tradition of behavior-based robotics (Brooks et al, 2004). The work of Connell is especially relevant, since it describes a mobile manipulator that uses behavior-based control to collect soda cans from the floor and tables (Connell, 1989).

2.4 Autonomous Grasping of Unmodeled Objects

Autonomous grasping has been an area of research since the dawn of computer-controlled robots (Ernst, 1962). The majority of these efforts have focused on grasping modeled objects, and the use of grasp planning. Methods capable of grasping unmodeled objects have been less common, but early examples do exist (Kamon et al, 1996; Sanz et al, 1998).

Recently, research has demonstrated that autonomous grasping of unmodeled objects is feasible via approaches that take advantage of compliant grippers, tactile sensing, and machine learning (Dollar and Howe, 2005; Natale and Torres-Jara, 2006; Saxena et al, 2008a,b). Our grasping approach is similar in spirit to this work. What distinguishes our approach is that it does not require training prior to being used, relies on object segmentation from 3D point clouds, and uses a straightforward grasp strategy. Unlike our system, most of this previous work does not address the problem of navigating up to the unmodeled object before grasping it.

2.5 Perception for Manipulation

EL-E uses a tilting laser range finder to acquire 3D point clouds around locations of interest. In contrast to model-based perception, EL-E's navigation and grasping behaviors segment parts of this point cloud, such as the surface and objects, and then transforms these segments into low-dimensional, task-relevant features.

Researchers have previously used visual segmentation to drive grasping, but these methods have typically relied on a uniform background and a stationary platform (Platt et al, 2005; Platt, 2006; Kamon et al, 1996; Dunn and Segen, 1988; Sanz et al, 1998). Our initial implementation of EL-E used a non-tilting laser range finder and an eye-in-hand camera to segment objects for grasping (Nguyen et al, 2008a).

Rusu et al (2008) present algorithms for combining multiple point clouds of indoor environments and creating semantic maps from this data. In contrast, our approach does not combine point clouds over time and only processes the points in a volume of interest (VOI) from which the robot extracts task-relevant features for a specific behavior.

2.6 Navigation for Manipulation

In our implementation, the robot navigates over relatively short distances (1.5m to 2m). We have divided the navigation into three distinct divisions detailed in Section 4.3. This is because the pose from which a range scan is taken significantly impacts the scan's usefulness due to the resolution of the point clouds and occlusion effects. For example, the scan must be taken very close (ca. 0.3m) to the object to segment small objects such as pills. Likewise, in order to reliably segment a surface using our method the scan must be made relatively close to the surface (ca. 0.6m).

There are prior examples of mobile manipulation systems that decompose the navigation task for similar reasons. MacKenzie and Arkin (1996) present a behavior-based system in which the robot can approach a drum and insert a probe into its bung hole. The bung hole is only visible when the robot is close to the drum, so it first navigates with respect to the drum, and then, once the bung hole is visible, navigates up to it and inserts the probe. Similarly, Platt et al (2006) use a three step turn-drive-turn control policy for a robot approaching and grasping a box. They show that this multi-step policy is beneficial, since the variance in the robot's estimate of the position of the box decreases as the robot gets closer.

3 The Robot's Body

The robot EL-E, is a statically stable mobile manipulator (shown in Figure 2) that consists of a 5-DoF Neuronics Katana 6M manipulator, an ERRATIC mobile base by Videre Design, and a 1-DoF linear actuator that can lift the manipulator and various sensors from ground level to 90cm above the ground (Nguyen et al, 2008a).

3.1 Exploiting a Symmetry

The degree of freedom provided by the linear actuator is critical to EL-E's ability to grasp and place objects on the floor and tables of different heights. EL-E uses its body to take advantage of a symmetry in the structure of indoor human environments by translating the manipulator and sensors to a canonical height with respect to the surface of interest. This enables it to use grasping and placing behaviors that are invariant to the height of the surface. Our experiments, described later, demonstrate that the robot performs well on flat surfaces of different heights. In contrast, the performance of the system is not invariant to other characteristics of the surface, such as how well the laser range finder segments objects sitting on it.

3.2 The Mobile Base, the Software, and the Sensors

The ERRATIC platform has differential drive steering with two powered wheels and one passive caster at the back. We replaced the motors of the ERRATIC platform with higher torque motors since we exceeded the payload of the standard motors. All computation is performed on-board with a single Mac Mini running Ubuntu GNU/Linux. We wrote most of our software in Python with occasional C++ and make use of a variety of open source packages including SciPy, Player/Stage, OpenCV and ROS (Robot Operating System) (Jones et al, 2001; Open Source Computer Vision Library: Reference Manual, 2001; Quigley et al, 2009).

For this work, EL-E uses three distinct types of sensors. First, it uses a laser pointer interface that consists of an omni-directional camera with a narrow-band green filter and a pan/tilt stereo camera. This detects a green laser spot and estimates its 3D location (Kemp et al, 2008).

Second, EL-E uses a laser range finder (Hokuyo UTM-30LX) mounted on a servo motor (Robotis Dynamixel RX-28) at the bottom of the aluminum carriage attached to the linear actuator. The servo motor tilts the laser range finder about the horizontal axis (Figure 3). We use this tilting laser range finder to obtain 3D point clouds of the environment. The laser range finder has a resolution of 0.25°and we obtain planar scans at 20Hz. The servo encoders have a resolution of 0.3°. All point clouds shown in this paper were obtained from this tilting laser range finder. The tilting laser range finder was inspired by a similar sensor on the Personal Robot 2 from Willow Garage (PR2 Technical Specs, 2008).

Several behaviors use 3D scans from this sensor, see Figure 5. The robot varies the angular range and rate of scanning depending on the task. For example, the grasping behavior uses a scan that results in a range image consisting of 320×200 3D points. This scan takes 24 seconds to acquire (due to averaging scan lines), has a vertical field of view (VFOV) of 80°, and a horizontal field of view (HFOV) of 60°. The behavior that navigates to the edge of the surface uses a 320×133 range image that takes 8 seconds to acquire (no averaging) with an 80° VFOV and a 40° HFOV.

Third, EL-E senses forces and torques using force-sensing fingers and a 6-axis force plate. We have replaced the Katana Sensor Fingers with our own custom fingers (Jain and Kemp, 2008). Each finger is a curved strip of aluminum covered with elastic foam for passive compliance and connected to the motor via a six-axis force/torque sensor (ATI Nano25 from ATI Industrial Automation). This enables us to measure the resultant forces and torques being applied to each finger independently. In addition to force sensing fingers, we have mounted the Katana on a 6-axis force plate (HE6X6 from AMTI). The force plate can sense a maximum force of 89N in the Z direction (vertical) with a resolution of approximately 0.044N, and sense 44N in the X and Y directions (shear) with a resolution of approximately 0.021N. In practice, noise lowers the effective resolution. The force plate allows the robot to sense forces applied to any point on the Katana arm. We currently use this force plate to detect collisions between the Katana and the environment.

4 Pick and Place System Description

4.1 Overview

Figure 5 shows a block diagram that illustrates the system. The input to the robot is a 3D location in the world defined with respect to the robot's coordinate frame. For this paper, the user provides this 3D location to the robot by briefly illuminating a location with a laser pointer. We have previously described how EL-E estimates the 3D coordinates for this illuminated location in an ego-centric frame with an average error of 10cm (Kemp et al, 2008). Although we use this laser pointer interface during the experiments in this paper, any method that provides EL-E with an appropriate 3D location would be sufficient to command the robot. For example, we have previously conducted tests with a touch screen interface (Choi et al, 2008). More generally, 3D locations could be provided by an autonomous perception system that selects locations based on the robot's goals.

The robot can perform two tasks, grasp an object from a flat surface and place an object already in its gripper on a flat surface. The specific task that the robot performs depends on the state of the robot (if it has an object in its gripper or not) and the local context around the 3D location selected by the user. For example, if the robot's gripper is free and the user selects an object on a table or the floor,

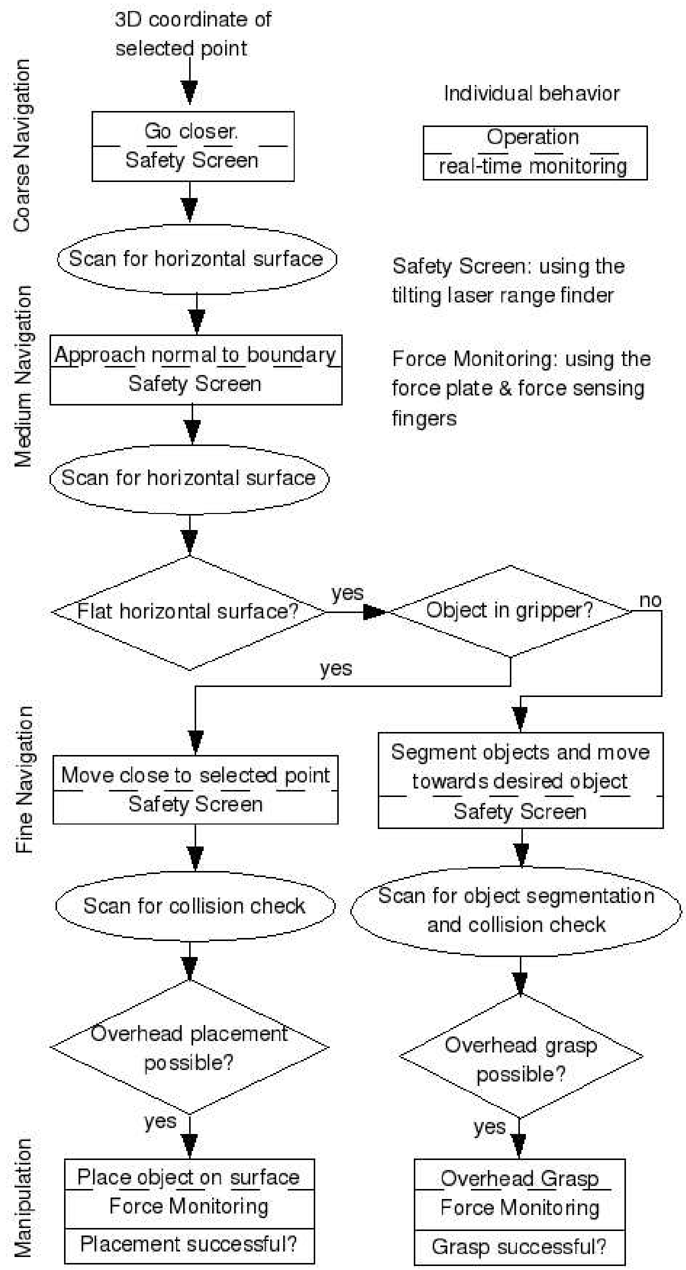


Fig. 5: This figure shows the different behaviors that the robot executes to grasp an object from a flat surface or place an object already in its gripper on a flat surface.

the robot will grasp it. If the user selects a door handle or an empty location on the floor, the robot will detect that the volume around the user-selected location does not have a flat surface and will not execute its grasping behavior.

The robot has real-time error detection for every behavior. While navigating, it uses the tilting laser range finder as a safety screen. It tilts the laser range finder to look down and in front for obstacle detection (Section 4.6.2). While

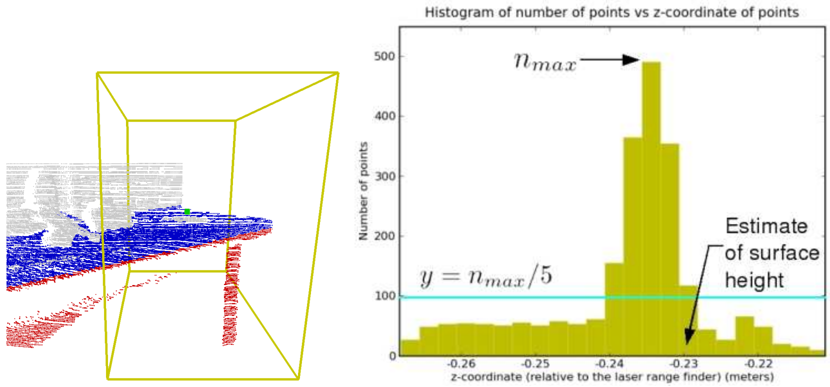


Fig. 6: (Color online) **Left:** This figure shows the output of the plane detection and segmentation algorithm. The grey, blue and red points are those that are above the surface, part of the surface and below the surface respectively. The yellow lines represent the boundary of the volume of interest around the user-selected location (green dot). **Right:** This histogram is for the part of the point cloud shown on the left that is inside the volume of interest (inside the yellow box). The X-axis is the z-coordinate of points relative to the laser range finder and the Y-axis is the number of points. Each bin represents a z-coordinate range of 2.5mm. The value of a bin is the number of points in the point cloud with a z-coordinate within the range of the bin. n_{max} is the value of the maximum bin and we use the line $y = n_{max}/5$ to find the top end of the surface as described in Section 4.2.1.

moving the manipulator, the robot monitors the force sensing fingers and the force plate and stops if it detects a collision with the environment. We assume that stopping in case of a collision or near collision is an appropriate behavior. Also, each behavior reports if it was completed successfully or not. For example, the grasp behavior reports if the grasp succeeded or not. If a behavior does not complete successfully, the robot falls back into a state where it waits for the user to select a location in the world.

4.1.1 Synopsis of a System Run Through

We now briefly describe the different steps that the robot goes through to grasp or place an object at the location selected by the user, see Figure 5. The robot uses the 3D coordinate of the selected location to move closer to the location, moves the laser range finder 20cm above the user-selected location and takes a 3D scan with the tilting laser range finder. It uses a subset of this scan around the user-selected location to detect a flat surface and approaches the user-selected location in a direction normal to the boundary of this surface (Sections 4.2.1 and 4.3).

Once the robot is close to the flat surface, it takes another 3D scan and uses that to refine its estimate of the height of the surface and to decide whether the user selected a point on a flat surface or not (Section 4.4.3). For this paper, if the user-selected location is not on a flat surface (e.g. the user selected a door handle, a light switch etc.), the robot does not execute the grasping or object placement behavior and goes back into a state in which it waits for the user to select a new location.

If the user-selected location is on a flat surface, the robot either places the object in its gripper on the surface, or attempts to grasp the object closest to the user-selected location with its empty gripper. The robot then navigates such that the user-selected location is within the workspace of its grasping or placing behavior. The robot makes another 3D scan to determine if the behavior can operate without a collision, and, in the case of grasping, segments the object to be grasped. Finally, the robot executes either the grasping or placing behavior.

4.2 Directed, Segmentation-Based Perception

Each of the behaviors starts with raw sensor data (point cloud consisting of $\sim$ 42,000 to 64,000 3D points) and reduces it to low-dimensional, task-relevant features (2 to 3 dimensional feature vectors). This reduction happens in three distinct steps.

First, each behavior has an associated volume of interest (VOI) that selects a subset of the point cloud. The robot focuses its perceptual resources on this spatially local volume and ignores the rest. Second, the behaviors use a mid-level representation that segments points in the VOI into a single flat surface and, possibly, a number of objects (usually fewer than 10). Third, each behavior transforms this mid-level representation into features that are specialized to the behavior's task (2 to 4 dimensions).

Below, we describe how the mid-level, segmentation-based representation is created.

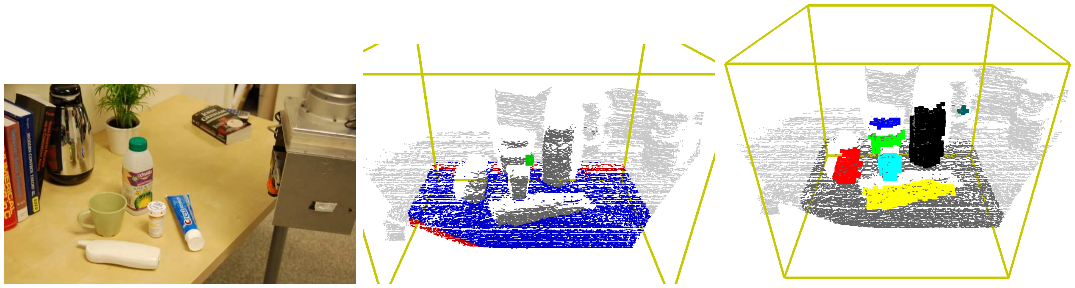


Fig. 7: (Color online) This figure shows the output of the segmentation algorithm. The left image shows the relative position of the actuated laser range finder and the objects. The middle image shows the output of the plane finding algorithm (red, blue and dark grey correspond to points below, on and above the flat surface). The segmented objects are shown in the image on the right. Points outside the volume of interest are shown in light grey.

4.2.1 Detecting and Segmenting the Surface

Both navigation and grasping rely on being able to detect and segment the surface (e.g., table top or counter top) relative to which the robot's actions will be performed. In this section we describe our algorithm for finding this surface in the volume of interest. The input to this algorithm is a 3D point cloud around the user-selected location (using the tilting laser range finder). Figure 6 shows the output of the algorithm. Our algorithm assumes that the surface is flat, approximately orthogonal to gravity within the volume of interest, and is represented by a significant number of points in the 3D point cloud.

To estimate the height of the flat surface within the volume of interest, the algorithm first quantizes the z-coordinate (parallel to gravity) within the volume of interest in intervals of 2.5mm and creates a histogram where the i^{th} bin represents a z-coordinate bound of $[z_i - 1.25 \times 10^{-3}, z_i + 1.25 \times 10^{-3})$ (z_i is the i^{th} quantized z-coordinate, in meters). The bin's value is the number of points in the point cloud that fall within the associated range of z. Figure 6 shows the point cloud from a 3D scan of a table and the histogram for the part of the point cloud within the VOI.

Let n_{max} be the maximum number of points in a single bin, bin_{max} be the bin with n_{max} points, and z_{max} be the quantized z-coordinate for bin_{max}. The surface segmentation algorithm assumes that bin_{max} is part of the flat surface. Starting at bin_{max} it searches over increasing height and looks for a bin with the number of points $< n_{max}/5$. The algorithm uses this as an estimate for the maximum height at which points will be classified as being part of the flat surface. The blue horizontal line in the histogram of Figure 6 is the line $y = n_{max}/5$.

Using a constant factor times n_{max} to decide which points to categorize as surface points is approximately the same as assuming that the distribution of the z-coordinate of points of the surface is Gaussian and moving a constant Mahalanobis distance up in height from the mean.

4.2.2 Segmenting Objects Sitting on the Surface

We now describe our algorithm to segment the objects sitting on the flat surface and within the volume of interest. The object segmentation algorithm first finds the surface (Section 4.2.1) and removes all points from the point cloud that are below or part of the surface.

It then converts the remaining point cloud into a 3D occupancy grid. The resolution of the grid is 1cm in the X and Y directions (parallel to the ground) and 0.25cm in the Z direction (vertical). The algorithm clusters the grid cells into objects by performing a 3D connected components labeling with 26-connectivity. This connected component labeling of the occupancy grid cells is comparable to agglomerative clustering of the point cloud above the surface.

It is worth noting that the assumptions about the surface play an important role in this process. If the algorithm did not remove the points associated with the surface, all of the objects would be connected together. The assumption about the existence of a surface upon which objects are sitting enables the segmentation algorithm to infer that the surface is distinct from the objects. Some uncommon situations will violate this assumption. For example, something may be bolted or glued onto a table. In these unlikely situations, the robot will detect a problem while trying to grasp the object using its force sensors.

After segmenting the objects, the algorithm finds the direction of maximum variance for each object and rejects objects with a length along the direction of maximum variance less than 1cm (noise from the flat surface) or greater than 50cm (too large to grasp). Note that this threshold of 1cm limits the ability of the robot to perceive especially small objects, such as pills. Figure 7 shows an example of the output from this segmentation algorithm.

4.3 Navigation

The navigation behaviors attempt to move the robot such that the grasping location for the object sitting on a surface is within the workspace of the grasping behavior (shown in Figure 8). Due to the resolution of the point clouds from the tilting laser range finder and occlusion effects, the pose from which a range scan is taken significantly impacts the scan's usefulness. For example, the scan must be taken very close (ca. 0.3m) to the object in order to segment small objects such as pills. Likewise, in order to reliably detect a surface the scan must be made relatively close to the surface (ca. 0.6 m).

At different times during the overall navigation process, EL-E navigates to the location selected by the user (*Coarse Navigation*, Section 4.3.1), navigates with respect to the boundary of the surface (*Medium Navigation*, Section 4.3.2), and navigates with respect to the object to be grasped (*Fine Navigation*, Section 4.3.3).

In each of these three cases, the navigation behavior reduces the raw sensory data from the world into a behavior-specific low-dimensional representation (a 3D location, a 3D location with an orientation, etc.). As it moves, EL-E uses wheel odometry to update the estimated pose of these static features in its ego-centric reference frame. This is sufficiently accurate, given the short distances over which EL-E currently navigates.

It is important to note that for this implementation each of these three navigation behaviors uses a simple turn, move straight ahead, turn method of navigation. However, any navigation method could potentially be substituted into each of these behaviors without altering the overall structure. For example, a path planar could be used to move the robot with respect to the user-selected location. The three divisions in navigation are not due to the way the robot moves, but rather result from the robot's sensing methods. For our current implementation, obtaining effective scans requires that the robot remain stationary. Even more significantly, the scans must be made from a favorable pose with respect to the volume of interest, so as to support the robot's next action.

4.3.1 Coarse Navigation: Navigation to the User-selected Location

The aim of *Coarse Navigation* is to move the robot to around 0.5m from the flat surface or 0.5m from the user-selected location, whichever condition stops the robot first. The robot turns to face the user-selected location, takes a 3D scan, removes points belonging to the ground plane, and then estimates how far forward it can go without colliding with an obstacle. Often this distance will correspond with how far the robot can go before it collides with the surface. The volume of interest in this case has a width and height equal to the width and height of the robot and extends forward until the user-selected location.

The robot moves in a straight line towards the user-selected location and stops when it is 0.5m away from the closest obstacle. If it stops around 0.5m from the surface of interest, then the surface detection and segmentation algorithm is more likely to work due to improved resolution.

Low-dimensional, Task-relevant Features: Coarse navigation uses the user-selected location, and the location of the closest obstacle.

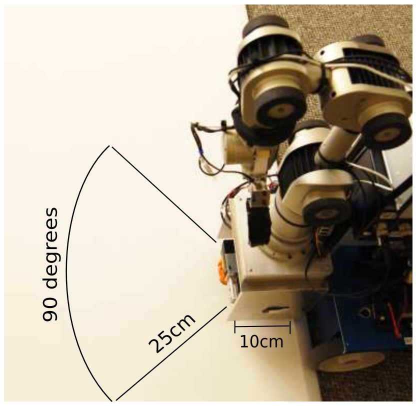
4.3.2 Medium Navigation: Navigation to the Edge of the Surface

After *Coarse Navigation*, the robot is close enough to the horizontal surface to segment the flat surface (Section 4.2.1) but is not close enough to the user-selected location to reliably segment objects of different sizes sitting on the flat surface. *Medium Navigation* approaches the user-selected location along a direction determined using a heuristic that tries to increase the chance that the object will be within the workspace of the grasping behavior.

Medium Navigation approaches the user-selected location along the vector between the user-selected location and the point on the front edge of the surface closest to the user-selected location. It assumes that the object is close to the user-selected location. This choice of approach direction achieves two goals. First, along this angular direction the user-selected location is in the center of the manipulator workspace, which increases the chance that the object will also be within the workspace. Second, the approach direction usually allows the robot to get close to the user-selected location without colliding with the edge of the table. This increases the chance that the robot will be able to get close enough to the user-selected location to place the object within the workspace of the grasping behavior.

Algorithm to Determine Approach Direction We now describe our algorithm to determine an approach direction (vector between the user-selected location and the point on the boundary of the surface that is closest to the user-selected location).

The first step is to segment out the flat surface. To do this, the algorithm first finds the height of the flat surface using the method described in Section 4.2.1. Let z_{top} be the z-coordinate of the top of the flat surface in the point cloud (in meters). The algorithm takes all the points whose z-coordinates lie in the range $[z_{top} - 0.02, z_{top}]$, projects them into 2D by removing the z-coordinate and then converts this



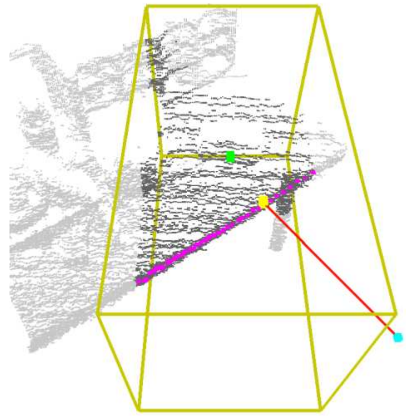


Fig. 8: (Color online) **Left:** This figure shows the workspace of the overhead grasp behavior. In the radial direction it ranges from 10cm to 35cm from the manipulator. The angular limits are imposed by the field of view of the laser range finder which is currently -45° and 45°, due to the design of the carriage. **Right:** This figure shows how the robot selects the direction to approach a flat surface. The green dot is the user-selected location and the cube defined by the yellow lines shows the volume of interest around the user-selected location. The dark grey points are those which are within the volume of interest. The pink dots are samples on the boundary of the surface, the yellow dot is the point on the boundary closest to the user-selected location and the approach direction is shown as a red line. The desired position of the robot (distance of 40cm from the yellow dot along the approach direction) is shown as a blue dot.

2D point cloud into a 2D occupancy grid. It finds objects in this grid using connected component labeling with 8-connectivity and selects the largest object (maximum number of cells) as the flat surface.

Then, the algorithm finds the points on the boundary of the surface using 2D binary erosion followed by subtraction. To find points which are in the line of sight of the robot, it first converts the boundary points into polar coordinates (r, θ). Then, it create bins which each represent an angular range of 1°and bins all the boundary points using their θ coordinate. For each bin, the algorithm selects the point with the minimum r as a representative point on the front edge of the surface. This gives a set of points on the front edge of the surface. The algorithm uses the point from this set that is closest to the user-selected location to determine the approach direction for the surface. Figure 8 shows the result of this algorithm on a 3D scan.

Since the algorithm determines the approach direction using discrete samples along the front edge, if the user-selected location is close to the front edge, there can be a significant error is the estimated approach direction. To avoid this, in the special case that the user-selected location is less that 5cm from the front edge of the surface, the algorithm uses a potential field from the samples on the front edge to push the user-selected location away from the front edge.

Moving to the Goal The robot moves in a straight line towards a point 40cm from the edge of the surface in the desired approach direction (blue dot in Figure 8). Once it reaches this point, it turns to face the user-selected location (tracked using wheel odometry). If the robot detects an obstacle in this straight-line path, it stops the straight line motion, turns to face the user-selected location and proceeds to the *Fine Navigation* behavior. We chose a distance of 40cm from the surface edge because it leaves enough space between the robot and the surface edge for the robot to rotate without danger of colliding with the surface.

Low-dimensional, Task-relevant Features: Medium Navigation uses a point 40cm from the edge of the surface in the desired approach direction and the user-selected location.

4.3.3 Fine Navigation: Navigation to the Object

The *Medium Navigation* behavior typically results in the robot being 40cm from the edge of the surface and facing normal to the surface's boundary. *Fine Navigation* attempts to move the robot such that the object is within the workspace of the manipulator.

The robot first moves closer to the user-selected location until it is 40cm from away from it (as opposed to 40cm from the edge of the surface – from *Medium Navigation*). The robot raises the tilting laser range finder 20cm above the surface, takes a 3D scan, and segments the objects sitting on the surface within the VOI around the user-selected location. It then selects the object closest to the user-selected location as the object to be grasped.

Let p_{close} be the location (2D coordinate) on the object to be grasped that is closest to the robot. The robot navigates such that the manipulator is 17cm from p_{close} and is oriented towards p_{close}. This is a heuristic to increase the

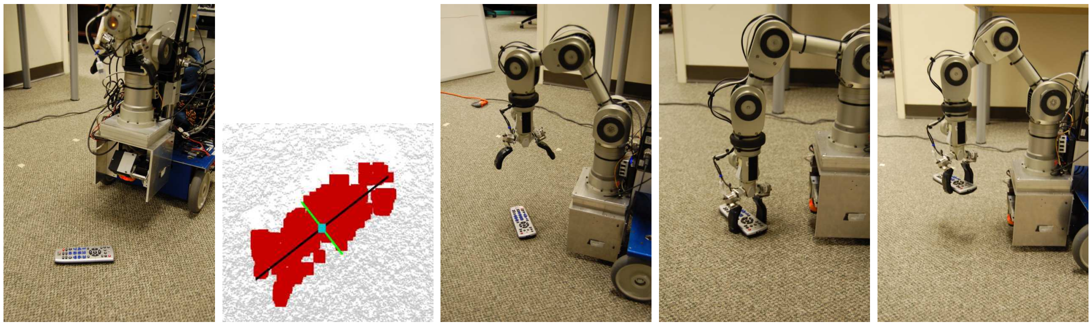


Fig. 9: (Color online) This figure shows the robot performing an overhead grasp to pick up a TV remote. It takes a 3D laser scan around the object and segments out the object. The blue dot in the second image is the centroid of the segmented object (red), the direction of maximum variance is the black line and the commanded orientation of the fingers is shown as the green line. The manipulator then moves the gripper above the object with the fingers oriented perpendicular to the direction of maximum variance. Finally, it descends down on the object, grasps it, and lifts it up.

chance that the grasp location on the object will be within the workspace of the manipulator (Figure 8). The robot now executes the grasping behavior.

If the object segmentation algorithm does not detect an object within the volume of interest, the robot navigates until it is 17cm from the user-selected location before executing the grasping behavior. The object (which we assume is close to the user-selected location) should now be closer to the robot, and the robot has another chance to segment the object in the grasping behavior.

At any time during this behavior, the robot's safety screen (Section 4.6.2) will detect if the robot is about to collide with the surface and stop the robot. Consequently, the robot moves close to its navigation goals without the risk of collision.

Low-dimensional, Task-relevant Features: Fine Navigation uses the user-selected location and a 2D location corresponding to the location on the object to be grasped that is closest to the robot.

4.4 Grasping

The navigation behaviors attempt to move the robot such that the object is within the workspace of the grasping behavior. The grasping behavior raises the tilting laser range finder above the surface on which the objects are sitting, takes a scan of the surface and the objects, segments the objects, attempts to find a suitable location and orientation for grasping, and tries to grasp the object. The grasping behavior does not move the mobile base, which ensures that the coordinate frame of the 3D point cloud and the manipulator are fixed relative to each other.

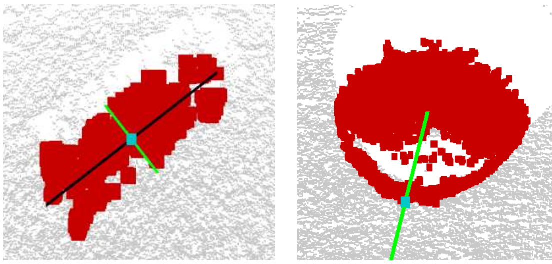
4.4.1 Overhead Grasp

The input to the overhead grasp behavior is the planar position of the gripper, which should place the gripper above the location to grasp, and the orientation of the wrist, whose rotational axis is parallel to gravity. The overhead grasp behavior assumes that the object to be grasped is sitting on the surface and has a height less than 20cm. The gripper starts at a constant height of 20cm above the estimated flat surface and descends until the robot detects contact or it has descended too far. Figure 9 shows the different stages that the robot goes through to perform an overhead grasp.

The robot first moves such that the gripper is directly above the grasping location and in the desired orientation. It then lowers the gripper down towards the object and the flat surface until either the IR sensor in the palm detects an object between the fingers, the force plate detects a contact between the manipulator and the object, or the force sensing fingers make contact with the surface (detected by a force greater than 1.0N in the vertical direction).

The robot then closes the gripper until it presses against the object or the surface, lifts up by a small amount and again closes the gripper. It repeats this close and raise action until either of the two fingers senses a force > 8N or the gripper closes to a minimum angle. We have chosen the minimum angle such that the gap between the two fingers is very small, but the fingers are not in contact with each other. This is important for detecting if the grasp was successful or not. The manipulator then moves the gripper up by 10cm and checks if it has an object in its grasp or not.

The robot uses the force sensing fingers to determine if there is an object in its grasp. If both the force sensing fingers sense a force less than 1.5N the overhead grasp behavior estimates that the gripper is not grasping an object and reports a grasping failure. As our tests show later in this paper, this method enabled EL-E to correctly detect



(a) TV remote (b) Bowl

Fig. 10: (Color online) This figure shows the output of the algorithm for selecting the position and orientation for grasping. The robot will try to grasp a TV remote at the centroid with the fingers oriented perpendicular to the direction of maximum variance (black line). For the bowl, it will try an overhead grasp at a raised point on the object that is close to the manipulator. In this case, it will orient the fingers along the line joining the grasp position and the centroid of the segmented object. The blue dot and green line indicate the commanded position of the gripper and orientation of the fingers respectively.

grasping success or failure with all objects except the dollar bill, which was too thin to result in a detectable force.

4.4.2 Deciding How to Grasp

The grasping behavior uses the point cloud associated with a segmented object to compute a low-dimensional, task-relevant feature vector (2D position and a 1D orientation) to command the overhead grasp.

After segmenting the object to be grasped using the algorithm described in Section 4.2.2, the grasping behavior creates a 2D binary image from the view-point of a virtual orthographic camera looking straight down at the object. This corresponds with the gripper's view-point during an overhead grasp. The binary image is constructed by converting the 3D points to 2D by removing the z-coordinate, and discretizing them using an occupancy grid in 2D (with a resolution of 1cm). The grasping behavior finds the centroid, the axes of maximum and minimum variance, and their associated variances for the object in this 2D binary image.

If the size of the object along the direction of minimum variance is less than the distance between the fingers of the gripper (12cm), the robot will attempt to grasp the object at the centroid of the 2D projection with the fingers oriented perpendicular to the axis of maximum variation.

Since the object is small enough (along the direction of minimum variation) to fit within the gripper, both the fingers should make contact with some part of the object as the gripper closes (assuming that the object slides on the flat surface). Grasping at the centroid is a rough approximation to grasping at the center of mass of an object, which can reduce the torque due to the weight of the object about the line joining the two fingers of the gripper. Orienting perpendicular to the direction of maximum variance also increases the chance that the fingers will make contact with the object given an error in the position of the gripper.

If the size of the object along the direction of minimum variance is greater than 12cm, the robot tries an overhead grasp at a raised point on the object that is close to the robot. The grasping behavior finds the point on the object with the maximum height above the flat surface. It then selects the location for the overhead grasp by finding the point closest to the manipulator from the set of points on the object which are less than 5mm below the point of maximum height. The grasping behavior determines the orientation of the fingers by the angle of the vector joining the centroid of the segmented object to the grasping location.

Figure 10 shows the output of the algorithm that decides how to grasp an object for a TV remote and a bowl. One can construct objects for which this method will not provide a successful location at which to grasp. However, as we demonstrate later, this method can succeed with important everyday objects.

Low-dimensional, Task-relevant Features: The grasping behavior uses the 2D centroid of the object, the axis of maximum variance, the axis of minimum variance, and the associated variances. If the object's minimum variance is greater than a threshold, it also uses a tall location on the object near the robot.

4.4.3 Preconditions for Grasping

The robot only executes the grasping behavior if it estimates that the following two pre-conditions are met. First, the volume of interest around the user-selected location must contain a flat surface (e.g., the user did not point at a door or a light switch). Second, the volume that the manipulator will sweep out during the grasping behavior must be empty.

A Flat Surface is Near the User-selected Location The robot assumes that the user has selected a location on a flat surface if the estimated flat surface within the volume of interest (VOI) occupies a large area parallel to the ground.

The robot finds the top of the estimated flat surface using the method described in Section 4.2.1. Let z_{top} be the z-coordinate of the top of the flat surface in the point cloud (in meters). The robot takes all the points whose z-coordinates lie in the range $[z_{top} - 0.1m,\ z_{top} + 0.1m]$, projects them into 2D by removing the z-coordinate, and then converts this 2D point cloud into a 2D occupancy grid. The width and height of this occupancy grid matches

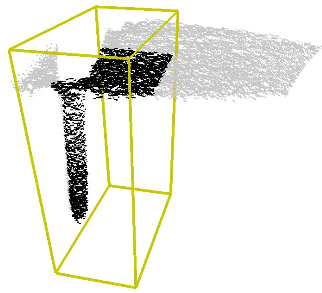


Fig. 11: (Color online) This figure shows the point cloud (grey and black) and an approximate volume that the manipulator will sweep out if it executes an overhead grasp (yellow). The points within the volume that the manipulator is likely to collide with are shown in black. In this example, the robot detects that an overhead grasp could result in a collision and does not execute the behavior.

a cross-section of the volume of interest parallel to the ground.

If the number of occupied cells in this 2D grid is greater than $1/4^{th}$ the total number of cells, the robot assumes that it is looking at a flat surface. This method of classification is similar to requiring that the area of the estimated flat surface within the VOI be greater than $1/4^{th}$ the area of cross-section of the volume of interest parallel to the ground, or greater than $400cm^2$.

Checking for an Overhead Collision Before attempting an overhead grasp, the robot checks if the trajectory for the grasp is likely to be collision free. It takes a scan of the region above and around the user-selected location, converts the point cloud into an occupancy grid for the volume that the manipulator will sweep out during the grasp and checks if this volume is empty. We approximate the volume that the manipulator will sweep out as a cuboid of height equal to the maximum height of the manipulator during an overhead grasp and width 20cm (the width of the open gripper). The length of the cuboid is the distance between the grasping point and the base of the manipulator plus 10cm (half the width of the open gripper).

Figure 11 shows the point cloud (grey points) and an approximate volume that the manipulator will sweep out (yellow cuboid) if the user-selected location is under a table. For this example, the intersection of the point cloud and the sweeping volume is not empty and the robot detects that an overhead grasp could result in a collision.

4.5 Object Placement

To place an object on a flat surface, the robot uses the same *Coarse Navigation* and *Medium Navigation* behaviors. In contrast to object grasping, the *Fine Navigation* behavior continues to approach the user-selected location rather than a segmented object. Also, if the user-selected location is less than 10cm behind the edge of the flat surface, the robot modifies the selected location's coordinates such that it is 10cm behind the edge of the table. This ensures that the robot does not place the object too close to the edge of the table, which can result in the object falling off.

EL-E then places the object at the (possibly modified) user-selected location. To place the object on the table, the manipulator moves the object above the desired point and then lowers it onto the surface. It detects the surface by a force ($> 2N$) in the vertical direction using the force sensing fingers. If the robot moves the object below the estimated height of the flat surface, and does not feel a force in the vertical direction, it detects an object placement failure. In this case, the robot will keep the object in its gripper and return to the state in which it waits for the user to select a location in the world.

Before executing the object placement behavior, the robot checks for the same pre-conditions as the grasping behavior. It checks for a flat surface around the user-selected location and a collision-free overhead placement using the methods described in Section 4.4.3.

Low-dimensional, Task-relevant Features: Object placement uses the user-selected location.

4.6 Real-time Monitoring

Each behavior has associated real-time monitoring to improve safety and detect errors. This is especially important in dynamic, unstructured environments, such as the home. The two most significant forms of monitoring involve using the laser range finder to form a safety screen that can detect when obstacles get close to EL-E's body, and using the force plate and force-sensing fingers to detect collisions with EL-E's arm.

4.6.1 Complementing Task-Relevant Sensing

Real-time monitoring complements EL-E's behaviors and enables EL-E to safely focus on the parts of the world that are most relevant to the current task. For example, when grasping an object, EL-E only pays attention to a small volume of the world centered around the object to be grasped. This implicitly makes the assumption that the rest of the world does not matter to this task. In other words, it assumes that the task is invariant to everything outside of this volume. This assumption helps EL-E generalize its capabilities to new environments, since only the contents of the volume need to meet EL-E's criteria for action.

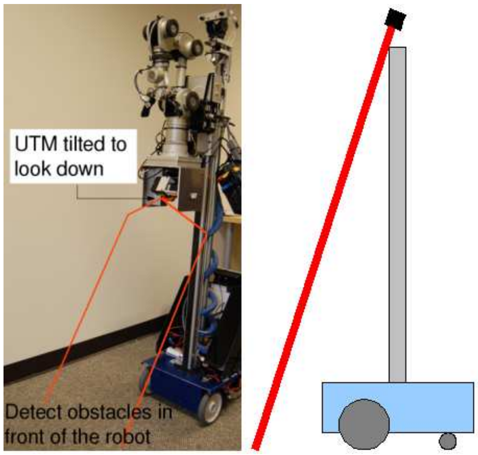


Fig. 12: The left image shows our current implementation using a laser range finder for real-time obstacle detection in the form of a safety screen. The figure on the right shows a sketch of our plans for a new actuated laser range finder that can pan and tilt in order to improve the coverage of the safety screen.

Of course, this assumption does not always hold true. For instance, a person could start out behind the robot, but interfere once the grasping behavior has started to move the arm. Since situations like this which violate EL-E's assumptions can happen, it is important to have real-time monitoring that can detect problems and signal for EL-E to stop and reassess the situation.

4.6.2 Obstacle Detection Using a Safety Screen

When EL-E moves, it lifts the actuated laser range finder to a height of approximately 90cm off the ground and tilts it down. In this way EL-E can detect obstacles, such as table tops and people, that get close to its body. This helps ensure that EL-E does not unintentionally collide with something. We refer to this as a safety screen. For future versions of EL-E, we plan to place an actuated laser range finder that can pan and tilt at the top of EL-E. This will enable EL-E to monitor for potential collisions over its entire body, regardless of the direction in which it is moving, see Figure 12. For now, EL-E lifts its carriage to approximate this sensor configuration.

As it moves, EL-E detects points in the laser scan above and below ground level. It can detect obstacles greater than 2cm in height. The height of the laser range finder (90cm) is greater than the height of the surfaces we used in our evaluation, including a relatively high counter top.

This method also allows the robot to detect if it tilts backwards (for example if it goes over a thick cable). Likewise it could be used to detect a descending flight of stairs as points in the scan below ground level. This method of obstacle detection is similar in spirit to the sensor configuration and obstacle detection algorithms used by teams in the DARPA Grand Challenge and Urban Challenge (Thrun et al, 2006; Urmson et al, 2008).

4.6.3 Collision Detection Using Force Sensing

The manipulation behaviors (grasping and placing) monitor the force plate and the force sensing fingers to detect collisions between the manipulator and the environment. The system reports a collision with the force sensing fingers when a change in the magnitude of the measured force exceeds a threshold (2.5N).

Detecting collisions with the force plate is harder because the forces and torques measured by the force plate depend on the configuration and motion of the manipulator. Rather than modeling the kinematics and dynamics of the arm explicitly, we use a data-driven approach, which is feasible because of the low-dimensional grasping behavior.

The overhead grasp and placement behaviors are parameterized by the 2D position and 1D orientation of the gripper. For this form of collision detection, the orientation of the gripper can be ignored leaving only two parameters. We performed a large number of collision-free overhead grasps covering the workspace of the grasping behavior and recorded the forces and torques for different configurations of the manipulator. Then, during normal operation, the robot looks up the stored configuration of the arm that is closest to the current arm configuration. It then find the difference between the collision-free forces and torques associated with this nearest neighbor configuration, and compares them to the currently measured forces and torques. If the difference is above a threshold of 4N or 2Nm, the robot declares a collision.

As we will revisit in the discussion, we believe this is an important example of the value of specialized behaviors that use low-dimensional, task-relevant parameters. It allows for efficient data-driven approaches and generally makes the robot more predictable.

5 Evaluation

In this section we evaluate the performance of the robot relative to variation in the environment and objects. First, we show the results of grasping two objects (a cordless phone and a pill) from 12 different flat surfaces (Section 5.1).

Then, in Section 5.2, we evaluate the performance of the robot in approaching and grasping objects from the 25 object categories that were ranked most important for robotic retrieval by motor-impaired patients from the Emory ALS Center (Choi et al, 2009b,c). We tested these objects on two surfaces, a carpeted floor and a table. We also present the performance of the grasp behavior in isolation for multiple trials with the same object on one surface.

We show results from an object grasping and placement experiment in Section 5.3. Finally, in Section 5.4, we analyze the common forms of failure to gain insight about the limitations of the current system.

5.1 Grasping Objects from 12 Different Surfaces

We evaluated the performance of the complete system on the following 12 distinct surfaces: two types of floor (carpeted and tiled), five different rectangular tables, a table with a curved edge, two circular tables, a kitchen counter top and the flat top of a set of drawers. We selected two objects, a cordless phone and a pill and performed two grasping trials for every surface, one for each object for a total of 24 trials on 12 different surfaces. We carried out these trials in three different rooms: the main room of the Healthcare Robotics Lab and a conference room and break room at the Health Systems Institute.

Table 1 shows the two objects, the 12 different flat surfaces and the performance of the robot. We started the robot around 1.5m away from the object in different starting orientations. A trial was deemed successful if the robot navigated to the surface and grasped the object. We varied the orientation of the cordless phone and the pill across the trials. We also varied the distance between the objects and the front edge of the flat surface between 5cm and 15cm.

We report the height of the different surfaces measured using a tape measure and the height estimated by the robot's surface detection algorithm. We present the results of these experiments in Table 1. The robot approached and grasped the phone from all 12 surfaces (100% success). It navigated to the pill and grasped it from 7 surfaces (58% success). For the five failures, the segmentation algorithm failed to segment the pill from the flat surface. The robot did navigate successfully to the user-selected location in the five failure cases. The surface detection and approach algorithms worked correctly in all 24 trials (100% success). The average of the absolute difference between the measured height and estimated height for the 24 trials was 7.1mm.

5.2 Grasping Multiple Objects from Two Flat Surfaces

We have proposed a prioritized list of 43 objects for the evaluation of assistive mobile manipulation systems operating in domestic settings (Choi et al, 2009b). We compiled this list by conducting a survey of 25 patients with ALS. Since this survey was contemporaneous, for this work we used a list of objects compiled from a survey of 15 patients, available as a technical report (Choi et al, 2009c).

Table 2: Multiple grasping trials for an object on one surface.

Object	Segmentation	Grasp
Pill	6/6 (100%)	6/6 (100%)
Plastic Fork	6/6 (100%)	4/6 (66.7%)

This table shows the result of grasping for multiple trials with a plastic fork and a pill on a table on which we could segment the pill (see Table 1). We placed the objects within the workspace of the grasping behavior, while varying their position and orientation.

We evaluated the performance of the robot on those objects which the ALS patients reported on average as being slightly important, important and very important for an assistive mobile manipulation system to be able to retrieve. This corresponds to the top 25 out of the ranked list of 43 everyday objects. An object from each of the 25 object categories is shown in Figure 13. The set of the top 25 object categories from the survey with 15 patients is identical to the survey of 25 patients, with the exception of credit card replacing straw in the list compiled from the survey of 25 patients.

We tested the navigation and overhead grasping behaviors for the floor and a table. For some categories, we performed trials with multiple representative objects. For example, we tested with transparent as well as opaque cups and bottles. We also performed multiple trials for some objects which had two sides with different characteristics (e.g. the cell-phone had one face that was shiny and one that was dull, the TV remote had one face that was black and the other metallic). As we discuss in Section 5.4, material properties play an important role in the success of our segmentation algorithm. We ran trials for the transparent cup and bottle, table knife, fork, spoon, keys, dollar bill, book, slipper and mail on one surface only (floor).

In each trial, the robot started around 1.5 meters away from the object. The experimenter briefly shined a green laser pointer on the object and the task for the robot was to navigate up to the object and pick it up. For objects with a well-defined major axis. The distance between the object and the front edge of the table was between 5cm and 15cm.

As shown in Figure 13, the robot grasped an object from every category at least once, except plates, books, table knives and mail. EL-E's grasping strategy can be successful on compliant objects, such as the hand towel and slipper. In contrast, these objects could be difficult to grasp using a traditional grasp planner that requires an explicit model of the object. Most grasp planning has focused on rigid objects.

Table 3 shows the results of the experiments for objects belonging to the 25 different categories. The third column

Table 1: Grasping two objects from 12 different surfaces.

						
	✓	✓	✓	✓	✓	✓
	✓	✓	×	✓	×	✓
Estimated height(m)	0.008 0.008	0.695 0.698	0.387 0.390	0.383 0.384	0.733 0.730	0.568 0.570
Measured height(m)	0.000	0.715	0.400	0.385	0.735	0.570
						
	✓	✓	✓	✓	✓	✓
	✓	✓	×	×	✓	×
Estimated height(m)	0.721 0.720	0.741 0.738	0.743 0.746	0.632 0.630	0.851 0.852	-0.002 -0.004
Measured height(m)	0.730	0.750	0.735	0.630	0.860	0.000

This table shows the performance of the robot in 24 trials for two objects on 12 different surfaces. A trial was successful if the robot navigated to the surface and grasped the object. The estimated height is the height of the surface as reported by the robot's surface detection algorithm for each trial. The first row reports the estimated height for trials with the phone and the second row is for trials with the pill. Measured height is the height we measured using a tape measure. The robot navigated to and grasped the phone from all 12 surfaces (100% success). It navigated to the pill and grasped it from 7 surfaces (58% success). For the five failures, the segmentation algorithm failed to segment the pill from the flat surface. The robot navigated successfully to the user-selected location in the five failure cases. The surface detection and approach algorithms worked correctly in all 24 trials (100% success). The average of the absolute difference between the measured height and estimated height for the 24 trials was 7.1mm.

shows that the overhead grasp behavior has a high success rate with objects that the segmentation algorithm can segment from the flat surface. The second column shows that the segmentation algorithm can fail on thin, transparent or reflective objects. We discuss the types of failure in detail in Section 5.4.

Table 2 shows the results of multiple grasping trials for two objects (a plastic fork and a pill) on one table. It shows that the grasping performance for a pill can be consistent on a given surface (6 successes out of 6 trials). In contrast, the robot grasped the fork successfully 4 out of 6 times, though the object segmentation algorithm returned a good position and orientation for all 6 trials. This happened because flat and thin objects can twist when the gripper applies a force on them, causing them to slip out of the two-fingered grasp.

5.3 Object Placement

To evaluate the object placement behavior we placed three objects (a TV remote, a toothbrush and a bowl) on a table and put tape on two tables to mark the desired object

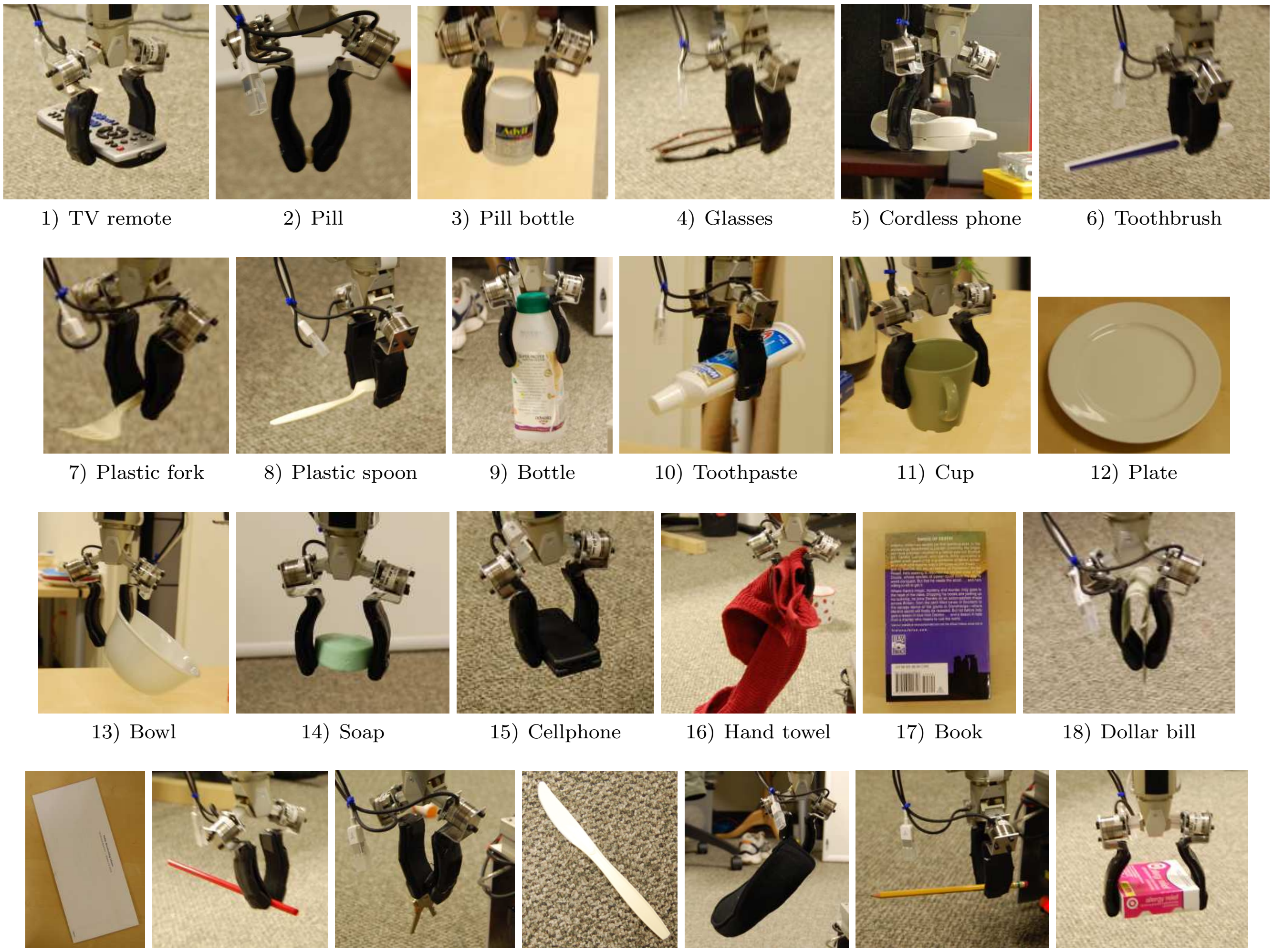


Fig. 13: This figure shows the 25 objects that we used to evaluate the performance of the robot. They are ordered by the importance ascribed to them by ALS patients (Choi et al, 2009c). The objects that the robot is shown grasping are the objects that the robot successfully navigated to, segmented, and grasped at least once during our experiments. Except for the fork, all of these photos were taken during the experiment described in Table 3. The fork picture is from an independent test of EL-E's ability to approach, segment, and grasp the fork. Although EL-E did not recognize that it had succeeded when grasping the dollar bill, we have counted it as a success here.

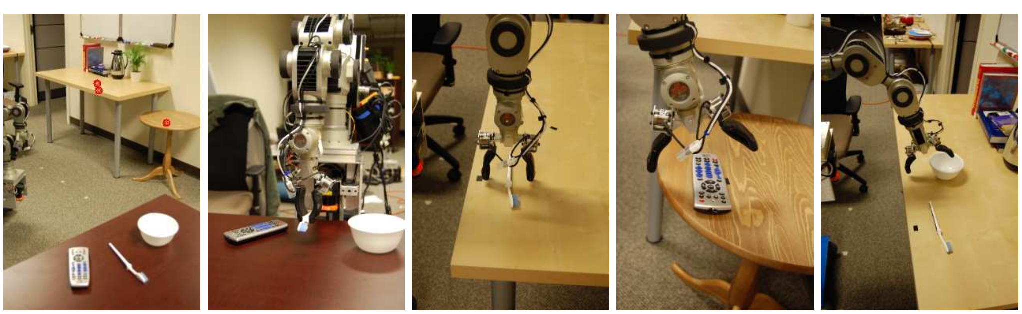


Fig. 14: (Color online) This figure shows the object placement experiment. The first image shows the three objects (TV remote, toothbrush and bowl) and the desired placement points (red circles). The second image shows the robot grasping the toothbrush, and the remaining three images show the robot placing the toothbrush, the TV remote, and the bowl.

Table 3: Grasping different objects from two surfaces.

Object	Approach	Object Segmentation	Grasp
Carpeted Floor			
TV remote (black face up)	✓	✓	✓
TV remote (metallic face up)	✓	✓	✓
Pill	✓	✓	✓
Pill bottle	✓	✓	✓
Glasses	✓	✓	✓
Cordless phone	✓	✓	✓
Toothbrush	✓	✓	✓
Plastic spoon	✓	✓	✓
Bottle	✓	✓	✓
Toothpaste	✓	✓	✓
Cup	✓	✓	✓
Bowl	✓	✓	✓
Soap	✓	✓	✓
Cellphone (dull face up)	✓	✓	✓
Cellphone (shiny face up)	✓	✓	✓
Hand towel	✓	✓	✓
Straw	✓	✓	✓
Keys	✓	✓	✓
Slipper	✓	✓	✓
Pencil	✓	✓	✓
Medicine box	✓	✓	✓
Slipper	✓	✓	×
Dollar bill	✓	✓	×
Plastic fork	✓	✓	×
Book	✓	✓	×
Mail	✓	×	
Table knife	✓	×	
Transparent Bottle	✓	×	
Transparent Cup	✓	×	
Plastic spoon	×		
	29/30 96.7%	25/29 86.2%	21/25 84.0%
Table			
TV remote (black face up)	✓	✓	✓
Pill bottle	✓	✓	✓
Cordless phone	✓	✓	✓
Bottle	✓	✓	✓
Toothpaste	✓	✓	✓
Cup	✓	✓	✓
Bowl	✓	✓	✓
Soap	✓	✓	✓
Cellphone (dull face up)	✓	✓	✓
Hand towel	✓	✓	✓
Medicine box	✓	✓	✓
TV remote (metallic face up)	✓	×	
Cellphone (shiny face up)	✓	×	
Straw	✓	×	
Toothbrush	✓	×	
Pencil	✓	×	
Pill	✓	×	
Glasses	✓	×	
	18/18 100%	11/18 61.1%	11/11 100%

This table shows the results of grasping experiments for the top 25 objects. Approach was deemed successful if the robot navigated such that the object was within the workspace of the grasping behavior. Object segmentation was deemed successful if the robot segmented out the desired object from the flat surface. A grasp was deemed successful if the robot picked up the object and detected that the grasp was successful.
We performed multiple trials for some objects which had two sides with different characteristics (e.g. the cell-phone had one face that was shiny and one that was dull, the TV remote had one face that was black and the other metallic). We ran trials for the transparent cup and bottle, table knife, fork, spoon, keys, dollar bill, book, slipper and mail on one surface only (floor).

placement locations. The experimental setup is shown in Figure 14. The task for the robot was to grasp the object selected using a laser pointer from the table and then place it at the desired location, which was also selected with the laser pointer. The robot successfully grasped and placed the three objects in series without error.

Figure 14 shows the robot placing the objects at the desired locations. The tape mark for the toothbrush was very close to the edge of the table, so the robot placed the object 10cm behind the front edge.

5.4 Failure Analysis

We will now look more carefully at the types of failures across the various tests we conducted.

5.4.1 Segmentation Failure

All of the behaviors rely on accurate detection and segmentation of the surface and the object of interest. We have found that this can fail for a variety of reasons.

Material Properties The laser range finder does not detect reflective objects such as metal spoons, forks and knives when viewed at an angle, and these objects tend to leave a hole in the point cloud. For transparent objects like glasses, transparent cups, and bottles the laser range finder returns an incorrect range. We believe that this is due to reflection and transmission of the IR laser. These perceptual failures cause the segmentation algorithm to either not detect the object at all, or to incorrectly estimate the object's position and orientation. Given these issues, we would expect for the robot to also perform poorly with surfaces made from comparable materials.

Geometric Properties The segmentation algorithm cannot segment flat and thin objects like mail and plastic knives that are sitting flat against the surface. Furthermore, point clouds for other small and thin objects vary from surface to surface. This is evidenced by the robot segmenting out objects like the pill, pencil, straw, glasses, plastic spoons, forks, and toothbrushes on the floor but not the table, resulting in the robot only succeeding on the floor.

We do not know the root cause of this difference, although it appears to relate to variations in the raw sensor readings. A qualitative inspection of the scans indicates that some small and thin object are not visible on some surfaces (e.g. the vitamin pill on some tables).

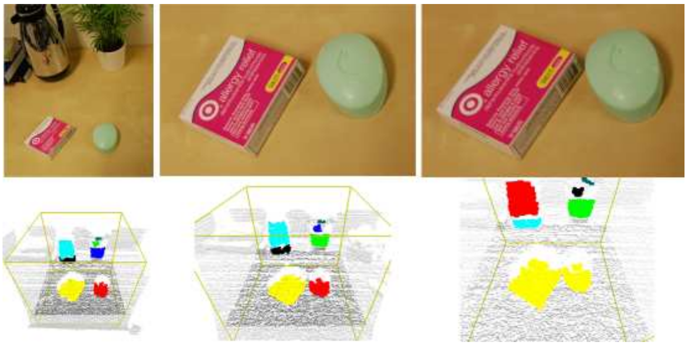


Fig. 15: (Color online) This figure shows the output of the object segmentation algorithm (bottom row) as the distance between the two objects is decreased from 6cm to 2cm to 1cm (left to right). When the distance between the objects equals the horizontal resolution of the occupancy grid (1cm) the segmentation algorithm merges the soap and the medicine box into one object (shown in yellow).

Clutter In Figure 15, we show an example of the performance of the object segmentation algorithm as the distance between two objects (medicine box and soap) is decreased from 6cm to 1cm. The object segmentation algorithm detects two objects when the distance between them is 6cm and 2cm. But, when we reduced the distance to 1cm (the horizontal resolution of the occupancy grid), the object segmentation algorithm merged the two objects into one. Extreme clutter can also interfere with the detection and segmentation of a surface, since these methods rely on many samples from the top of the 3D surface and clutter can reduce the number of samples due to occlusion.

5.4.2 Grasping

A second type of failure occurred due to the grasping strategy. The robot cannot grasp objects like books and plates that are too large to fit within the two fingered gripper and do not have an edge that the robot can grasp (like bowls). Thin and rigid objects like spoons and forks can sometimes twist and slip out of the grasp when the fingers apply a force on them. Finally, though the robot could segment and grasp a used dollar bill from the floor in the configurations we tested, it did not detect there was a dollar bill in its gripper and thus it declared a grasp failure.

We did not formally test the performance of the system in clutter beyond the crowding shown in the object placement test. The mechanical impact of clutter is exacerbated by the current system, since EL-E does not pre-shape its fingers. Instead, it always descends down on an object with the gripper open as wide as possible. We expect that without sufficient distance between the objects, the overhead grasp behavior would descend and collide with objects other than the object of interest.

6 Discussion

Currently, there are no clear answers about how to create general purpose autonomous mobile manipulators that perform useful tasks in dynamic, unstructured settings. Our approach focuses on the use of specialized behaviors that make use of low-dimensional, task-relevant features. This is similar in spirit to work by other researchers (Katz et al, 2008; Ciocarlie et al, 2007; Jenkins, 2008).

Biological systems offer very good examples of autonomous mobile manipulators. Insects and animals often use specialized perceptual features as input to specialized behaviors (Goodenough et al, 1993; N. Cowan, 2006), and evidence indicates that some seemingly complex human actions may be explained by control in a low-dimensional task space derived from high-dimensional sensor and muscle spaces (Ting, 2007). Although success in biological control does not imply that robots should use similar methods, we believe that similar methods offer many benefits to robots.

First, ignoring irrelevant sensory information enables a robot to generalize its capabilities to new situations and focus its perceptual resources on what is important. For example, EL-E is able to operate in diverse environments partly because each of its behaviors only focuses on a volume of interest (VOI) tailored to its needs. In our current implementation, the VOI is centered around a 3D location provided by the human user, but it could plausibly be provided by an autonomous perception system, such as those found in Meger et al (2008) and Rusu et al (2008).

Second, due to their low complexity, specialized behaviors are more predictable and easier to characterize and monitor for unexpected situations. For example, the low-dimensional parameterization of the overhead grasp behavior enables the robot to easily monitor for collisions, see Section 4.6.3. Also, we were able to easily characterize the workspace of the overhead grasp behavior so that it could be composed with the navigation behavior.

Third, the use of specialized behaviors naturally leads to modularity and extensibility. Although the overhead grasp behavior works well for objects sitting on flat surfaces that are unobstructed from above, there are situations where it would not be suitable, such as grasping objects from a shelf, grasping large objects like books and plates, and thin and flat objects such as mail and dollar bills. We have shown that it is relatively straightforward to select between two types of grasp locations (e.g., middle of an object versus the edge of a bowl), and to define pre-conditions to determine whether the overhead grasp behavior is appropriate or not. We believe that the grasping behavior can be extended by adding specialized behaviors that grasp objects from the front, grasp thin objects, and handle other specialized cases not covered by the existing behaviors. An appropriate behavior could then be selected based on various feature detectors and pre-conditions. We believe that by using a few specialized behaviors, the robot can handle large variations in the environment. We hope to explore this in future work. In order to expand EL-E's capabilities beyond object grasping, we have already developed additional behaviors that deliver an object to a person or open a door (Choi et al, 2009a; Jain and Kemp, 2008).

Another point we wish to emphasize is that relatively simple behaviors can sometimes perform well over a large set of situations. A simple behavior and mechanism does not imply narrow functionality relative to the common situations found in the real world. We believe the performance of our grasping behavior provides evidence of this, and also points to the importance of strong empiricism. At this time, only experiments in the real world can convincingly test the performance of an autonomous mobile manipulator with respect to true task variation. Without building and testing complete systems, we are unlikely to be able to determine where the most pressing challenges lie.

Clearly, there are still significant limitations to our system. For example, our grasping behavior would be oblivious to the risks of grasping a cup filled with liquid, and the robot only navigates over short distances. Nonetheless, we believe that these issues can be handled by building on the current system and that the use of behaviors with pre-conditions, real-time monitoring, and post-conditions will help keep the system extensible (by post-conditions we mean that a behavior can detect whether it was successful or not). Post-conditions are especially important, because we believe that a key to robustness is knowing when to try again, possibly with a different behavior.

7 Conclusion

We have presented our progress towards the creation of an assistive robot that can autonomously pick and place objects in the home. Significant challenges remain, including navigating and grasping within real homes, which are likely to have significant clutter, overcoming sensor limitations, operating at faster speeds, increasing overall reliability, and expanding the set of objects that can be grasped. We hope to address these issues in future work.

Acknowledgements We would like to thank the members of the Healthcare Robotics Lab at Georgia Tech, and everyone who has made contributions to EL-E, including Cressel Anderson, Hai Nguyen, Young Sang Choi, Zhe (Joseph) Xu, Travis Deyle, Tiffany Chen, Alex Trevor, Dr. Jonathan Glass, and the wonderful patients and staff at the Emory ALS Clinic. This research was supported in part by Willow Garage and NSF grant IIS-0705130.

References

Asfour T, Azad P, Vahrenkamp N, Regenstein K, Bierbaum A, Welke K, Schröder J, Dillmann R (2007) Toward humanoid manipulation in human-centred environments. Robotics and Autonomous Systems

Bluethmann W, Ambrose R, Fagg A, Rosenstein M, Platt R, Grupen R, Brezeal C, Brooks A, Lockerd A, Peters R, Jenkins O, Mataric M, Bugajska M (2004) Building an autonomous humanoid tool user. In: International Conference on Humanoid Robots

Brock O, Khatib O (2002) Elastic strips: A framework for motion generation in human environments. The International Journal of Robotics Research

Brooks R, Aryananda L, Edsinger A, Fitzpatrick P, Kemp C, O'Reilly UM, Torres-Jara E, Varshavskaya P, Weber J (2004) Sensing and manipulating built-for-human environments. International Journal of Humanoid Robotics

Choi YS, Anderson CD, Glass JD, Kemp CC (2008) Laser pointers and a touch screen: Intuitive interfaces to an autonomous mobile robot for the motor impaired. In: ACM SIGACCESS Conference on Computers and Accessibility

Choi YS, Chen T, Jain A, Anderson C, Glass JD, Kemp CC (2009a) Hand it over or set it down: Object delivery with an assistive mobile manipulator. In: Proceedings of the IEEE 18th International Symposium on Robot and Human Interactive Communication (RO-MAN) – To Appear

Choi YS, Deyle T, Chen T, Glass JD, Kemp CC (2009b) Benchmarking Assistive Mobile Manipulation: A List of Household Objects for Robotic Retrieval Prioritized by People with ALS. In: International Conference on Rehabilitation Robotics

Choi YS, Deyle T, Kemp CC (2009c) A list of household objects for robotic retrieval prioritized by people with als. Tech. Rep. 0902.2186v1, URL http://arxiv.org/abs/0902.2186

Ciocarlie M, Goldfeder C, Allen P (2007) Dimensionality reduction for hand-independent dexterous robotic grasping. In: International Conference on Intelligent Robots and Systems (IROS)

Connell J (1989) A behavior-based arm controller. IEEE Transactions on Robotics and Automation

Dallaway J, Jackson R (1992) RAID a Vocational robotic workstation. In: Proceedings of the 1992 International Conference on Rehabilitation Robotics.

Dario P (1999) MOVAID: a personal robot in everyday life of disabled and elderly people. Technology and Disability

Dollar AM, Howe RD (2005) Towards grasping in unstructured environments: Grasper compliance and configuraton optimization. Advanced Robotics

Dunn G, Segen J (1988) Automatic discovery of robotic grasp configurations. In: 1988 IEEE International Conference on Robotics and Automation

Ernst H (1962) MH-1, a computer-operated mechanical hand. PhD thesis, Massachusetts Institute of Technology

Goodenough J, McGuire B, Wallace R (1993) Perspectives on animal behavior. J. Wiley New York

Graf B, Hans M, Schraft RD (2004) Care-O-bot II - Development of a next generation robotic home assistant. Autonomous Robots

Grupen R, Brock O (2004) White paper: Integrating manual dexterity with mobility for human-scale service robotics - the case for concentrated research into science and technology supporting next-generation robotic assistants

Hernandez M, Echeverry S, Guerra R, Kim D, Hamilton C, Behal A (2008) User-Guided Vision-Based Control of a Wheelchair Mounted Assistive Robotic Arm in an Unstructured Environment. In: Proc. of RESNA

Hillenbrand U, Ott C, Brunner B, Borst C, Hirzinger G (2004) Towards service robots for the human environment: the Robutler. In: Proceedings Mechatronics & Robotics (MechRob)

Inamura T, Okada K, Tokutsu S, Hatao N, Inaba M, Inoue H (2008) HRP-2W: A humanoid platform for research on support behavior in daily life environments. Robotics and Autonomous Systems

Jain A, Kemp CC (2008) Behaviors for robust door opening and doorway traversal with a force-sensing mobile manipulator. In: RSS Workshop on Robot Manipulation: Intelligence in Human Environments

Jenkins OC (2008) Sparse control for high-dof assistive robots. Intelligent Service Robotics 1(2):123–134

Jones E, Oliphant T, Peterson P, et al (2001) SciPy: Open source scientific tools for Python. URL http://www.scipy.org/

Kamon I, Flash T, Edelman S (1996) Learning to grasp using visual information. In: IEEE International Conference on Robotics and Automation

Katz D, Kenney J, Brock O (2008) How Can Robots Succeed in Unstructured Environments? In: RSS Workshop on Robot Manipulation: Intelligence in Human Environments

Kemp CC, Edsinger A, Torres-Jara E (2007) Challenges for robot manipulation in human environments. IEEE Robotics & Automation Magazine

Kemp CC, Anderson CD, Nguyen H, Trevor AJ, Xu Z (2008) A point-and-click interface for the real world: Laser designation of objects for mobile manipulation. In: International Conference on Human-Robot Interaction

Khatib O, Yokoi K, Brock O, Chang K, Casal A (1999) Robots in human environments: Basic autonomous capabilities. International Journal of Robotics Research

Kragić D, Petersson L, Christensen H (2002) Visually guided manipulation tasks. Robotics and Autonomous Systems

Kwee H, Duimel J, Smits J, de Moed A, van Woerden J (1989) The MANUS Wheelchair-Borne Manipulator: System Review and First Results. In: Proc. IARP Workshop on Domestic and Medical & Healthcare Robotics, Newcastle

Van der Loos H (1995) VA/Stanford rehabilitation robotics research and developmentprogram: lessons learned in the application of robotics technology tothe field of rehabilitation. IEEE Transactions on Rehabilitation Engineering 3(1):46–55

Van der Loos H, Wagner J, Smaby N, Chang K, Madrigal O, Leifer L, Khatib O (1999) ProVAR assistive robot system architecture. In: IEEE International Conference on Robotics and Automation.

Van der Loos M, Reinkensmeyer DJ (2008) Springer Handbook of Robotics, Springer, chap 53 Rehabilitation and Health Care Robotics

MacKenzie D, Arkin R (1996) Behavior-based mobile manipulation for drum sampling. In: IEEE International Conference on Robotics and Automation

Meger D, Forssen P, Lai K, Helmer S, McCann S, Southey T, Baumann M, Little J, Lowe D, Dow B (2008) Curious george: An attentive semantic robot. Robotics and Autonomous Systems

N Cowan RF J Lee (2006) Task-level control of a rapid wall following in the american cockroach. Journal of Experimental Biology

Natale L, Torres-Jara E (2006) A sensitive approach to grasping. In: Proceedings of the Sixth International Workshop on Epigenetic Robotics

Nguyen H, Anderson CD, Trevor AJ, Jain A, Xu Z, Kemp CC (2008a) EL-E: An Assistive Robot that Fetches objects from Flat Surfaces. In: Robotic Helpers, Int. Conf. on Human-Robot Interaction

Nguyen H, Jain A, Anderson CD, Kemp CC (2008b) A clickable world: Behavior selection through pointing and context for mobile manipulation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Okada K, Ogura T, Haneda A, Fujimoto J, Gravot F, Inaba M (2005) Humanoid motion generation system on HRP2-JSK for daily life environment. In: IEEE International Conference Mechatronics and Automation

Open Source Computer Vision Library: Reference Manual (2001) Open source computer vision library: Reference manual. URL http://sourceforge.net/projects/opencvlibrary

Platt R (2006) Learning and generalizing control-based grasping and manipulation skills. PhD thesis, University Massachusetts

Platt R, Fagg A, Grupen R (2005) Re-using schematic grasping policies. IEEE International Conference on Humanoid Robots

Platt R, Burridge R, Diftler M, Graf J, Goza M, Huber E, Brock O (2006) Humanoid mobile manipulation using controller refinement. In: IEEE International Conference on Humanoid Robots

PR2 Technical Specs (2008) URL www.willowgarage.com/pages/robots/technical-specs

Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J, Eric Berger RW, Ng A (2009) ROS: an open-source Robot Operating System. In: Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA)

Remazeilles A, Leroux C, Chalubert G (2008) SAM: A robotic butler for handicapped people. In: IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)

Rusu R, Marton Z, Blodow N, Dolha M, Beetz M (2008) Towards 3D Point cloud based object maps for household environments. Robotics and Autonomous Systems

Sanz P, Del Pobil A, Iñesta J, Recatala G (1998) Vision-guided grasping of unknown objects for service robots. In: IEEE International Conference on Robotics and Automation

Saxena A, Driemeyer J, Ng A (2008a) Robotic Grasping of Novel Objects using Vision. The International Journal of Robotics Research

Saxena A, Wong L, Ng A (2008b) Learning Grasp Strategies with Partial Shape Information. In: AAAI

Srinivasa S, Ferguson D, Weghe M, Diankov R, Berenson D, Helfrich C, Strasdat H (2008) The Robotic Busboy: Steps Towards Developing a Mobile Robotic Home Assistant. In: International Conference on Intelligent Autonomous Systems

Stanger CA, Anglin C, Harwin WS, Romilly DP (1994) Devices for assisting manipulation: a summary of user task priorities. IEEE Transactions on Rehabilitation Engineering

Thrun S, Montemerlo M, Dahlkamp H, Stavens D, Aron A, Diebel J, Fong P, Gale J, Halpenny M, Hoffmann G, et al (2006) Stanley: The robot that won the DARPA Grand Challenge. Journal of Field Robotics

Ting L (2007) Dimensional reduction in sensorimotor systems: a framework for understanding muscle coordination of posture. Computational Neuroscience: Theoretical Insights Into Brain Function

Topping M, Smith J (1998) The development of Handy 1, a rehabilitation robotic system to assist the severely disabled. Industrial Robot

Tsui K, Yanco H, Kontak D, Beliveau L (2008) Development and evaluation of a flexible interface for a wheelchair mounted robotic arm. In: Proceedings of the 3rd international conference on Human Robot Interaction

Urmson C, Anhalt J, Bagnell D, Baker C, Bittner R, Clark M, Dolan J, Duggins D, Galatali T, Geyer C, et al (2008) Autonomous driving in urban environments: Boss and the Urban Challenge. Journal of Field Robotics

Waarsing B, Nuttin M, Brussel HV (2001) Introducing robots into a human-centred environment - the behaviour-based approach. In: International Conference on CLAWAR

Zllner R, Asfour T, Dillmann R (2004) Programming by Demonstration: Dual-Arm Manipulation Tasks for Humanoid Robots. In: International Conference on Intelligent Robots and Systems

Advait Jain is a Ph.D. student at the Georgia Institute of Technology in the School of Interactive Computing. He is a member of the Center for Robotics and Intelligent Machines at Georgia Tech (RIM@GT), the Health Systems Institute, and the Healthcare Robotics Lab. He received a Bachelor of Technology degree in Computer Science and Engineering from the Indian Institute of Technology (IIT), Delhi, India, in 2007.

Charles C. Kemp is an Assistant Professor at the Georgia Institute of Technology in the Wallace H. Coulter Department of Biomedical Engineering at the Georgia Institute of Technology and Emory University. He is also an Adjunct Assistant Professor in the School of Interactive Computing. He received a Ph.D. in Electrical Engineering and Computer Science (EECS) from the Massachusetts Institute of Technology (MIT) in 2005, an M.Eng. in EECS from MIT in 1998, and a B.S. in Computer Science and Engineering from MIT in 1997. He is a member of the Center for Robotics and Intelligent Machines at Georgia Tech (RIM@GT) and the Health Systems Institute, which houses his lab, the Healthcare Robotics Lab. Charlie's research interests include autonomous mobile manipulation, human-robot interaction, assistive robotics, healthcare robotics, bio-inspired approaches to robotics, and AI.