Strategies for Human-in-the-Loop Robotic Grasping

Adam Leeper, Kaijen Hsiao,
Matei Ciocarlie, Leila Takayama
Willow Garage, Inc.
68 Willow Road, Menlo Park, CA 94025, USA
{aleeper, hsiao, matei,
takayama}@willowgarage.com

David Gossow
Technische Universität München
Department of Computer Science
Informatik 9, Karlstraße 45, 80333 Munich
david.gossow@cs.tum.edu

ABSTRACT

Human-in-the loop robotic systems have the potential to handle complex tasks in unstructured environments, by combining the cognitive skills of a human operator with autonomous tools and behaviors. Along these lines, we present a system for remote human-in-the-loop grasp execution. An operator uses a computer interface to visualize a physical robot and its surroundings, and a point-and-click mouse interface to command the robot. We implemented and analyzed four different strategies for performing grasping tasks, ranging from direct, real-time operator control of the end-effector pose, to autonomous motion and grasp planning that is simply adjusted or confirmed by the operator. Our controlled experiment (N=48) results indicate that people were able to successfully grasp more objects and caused fewer unwanted collisions when using the strategies with more autonomous assistance. We used an untethered robot over wireless communications, making our strategies applicable for remote, human-in-the-loop robotic applications.

Categories and Subject Descriptors

I.2.9. [**Robotics**]: Operator Interfaces; H.1.2. [**User/Machine Systems**]: Human Factors

General Terms

Design, Experimentation, Human Factors, Performance

Keywords

grasping, teleoperation, shared autonomy

1. INTRODUCTION

As personal service robots make progress towards performing daily tasks in the home and office, they must be able to deal with complex and changing environments. Despite significant progress in many related fields, execution of a complex task still poses significant challenges. One option for accelerating the development of service robots is to involve a human in the loop. An operator's cognitive abilities can be tapped to deal with corner conditions and decisions that are difficult for autonomous systems. Such a system could be reliable enough for real-world deployment in the near future, making personal service robots a possible solution to the expected increasing cost and shortage of unskilled manual labor.

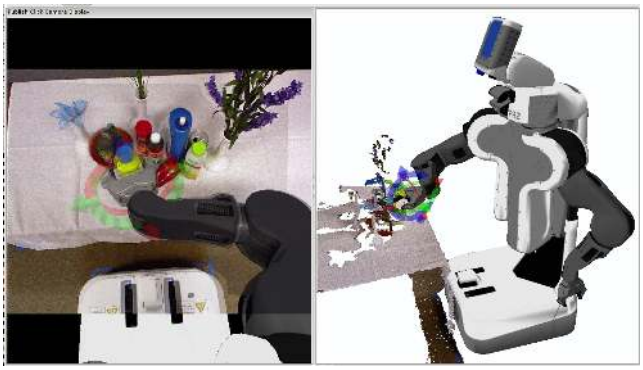


Figure 1: A remotely operated robot performing a grasping task, as seen through the point-and-click interface used by the operator.

In a Human-in-the-Loop (HitL) framework, autonomous capabilities can reduce the load on the operator and increase overall task efficiency. While a robot might not be able to autonomously handle a complete task in a robust and general way, we can use autonomy for sub-tasks that can be performed reliably, or that require operator input in a form that is relatively effortless to provide. Identifying and building such techniques, studying their interplay with operator-controlled sub-tasks, and analyzing the overall gain in efficiency are all steps towards deployable HitL systems.

In this paper, we focus on the ability to grasp and hold objects in the robot end-effector, a key prerequisite for many household and office tasks involving object transport and manipulation. A complete grasping task involves choosing an end-effector pose relative to the object, and executing an arm trajectory to bring the end-effector there. The robot must achieve a stable grasp of the object while avoiding undesired collisions with other parts of the environment. This can be challenging for the operator, particularly when dealing with non-anthropomorphic arms with many degrees of freedom (DOFs) or limited sensor data of the environment.

HRI'12, March 5–8, 2012, Boston, Massachusetts, USA.

To alleviate some of these difficulties, we investigate four strategies for HitL grasp execution. These strategies, which will be presented in detail in Sec. 4, can be summarized as follows:

- **Strategy 1: Direct control**. Operator directly controls the 6D pose of the gripper in real-time.
- **Strategy 2: Waypoint following**. Operator specifies desired gripper position (waypoint) goals and adjusts them until satisfied before asking the robot to move.
- **Strategy 3: Grasp execution**. Operator only specifies the final desired grasp pose; robot performs collision-free motion planning to execute the specified grasp.
- **Strategy 4: Grasp planning**. Operator indicates general area for grasping; robot computes grasp pose suggestions for the operator to select and optionally adjust.

We aim to study performance in unstructured human environments. In particular, we test on objects previously unknown to the robot, immersed in clutter and/or surrounded by obstacles; only 2D images and noisy 3D point clouds of the scene are available for both grasp and motion planning.

The main contribution of our work is twofold. On one side, we present a HitL robotic system for efficient grasp execution in complex, unstructured environments. More importantly, we explore how operator input can be combined with autonomous methods for increasing task efficiency in the context of a grasping task. We believe that the comparison and analysis of such strategies can help guide the design of future HitL systems for more complex manipulation tasks.

2. RELATED WORK

A variety of teleoperation interfaces for robot arms have been used in applications such as space and undersea robotics, rescue robotics, and robotic surgery. Shared, supervisory, and collaborative control all refer to using a combination of autonomous and human control. Early investigations in shared control focused on the shared execution of motion trajectories with moderate time delays, as encountered in space [6] [14]. Towards dexterous manipulation, studies have used shared control for task primitives such as grasping and peg insertion [13]. For systems with many degrees of freedom, virtual fixtures [1] and other forms of haptic assistance in surgical applications [15] help to constrain motions of the master. There are numerous other types of assistance that a shared control interface can provide, including various degrees of supervisory control [19], autonomous assistance through mixed initiative user interfaces [8], interface agents [11], and other intelligent user interfaces [12].

The teleoperation literature has shown that many factors can affect the performance of a remote teleoperator. Relevant to this work are the issues of viewpoint, depth perception, and time delay. Our interface provides two views for the user: a monocular view from a camera on the robot's head; and a virtual, 3D rendering of the robot together with the color point cloud seen by the robot's sensors. This is a similar strategy to [7, 18], however, those interfaces require full models of the scene and objects, which are not available in our task. The viewpoint in both of our views is adjustable (the robot head can be pivoted to look at different objects, while the virtual 3D-view is free-roaming), which is driven by prior work showing that free-roaming viewpoints are superior to fixed views for manipulation tasks [3].

Another study comparing different interfaces for teleoperating robot arms showed that providing operators with 6D movement can be beneficial even if fewer DOFs are required for the task [16]. In [20], motion planning was shown to be useful in a comparison of five teleoperation strategies that bring a point on a simulated 6DOF robot's end-effector to a 3D positional goal in space. Our task also includes commanding a robot arm using a mouse, and part of it requires moving through free space while avoiding obstacles. However, grasping inherently requires controlling end-effector orientation as well as position, and also requires goal (grasp) selection as well as contact with objects in the environment during execution.

In the long history of teleoperation, a variety of models have been defined to describe the control strategies in human-robot task completion. To help frame our work within this context, we describe some of these models for teleoperation:

- Direct control, where the user directly controls the motion of the robot, with no intelligence or autonomy.
- Shared control, where the robot controls some aspects of a task, while the user still controls low-level motions [4].
- Supervisory control, where a user issues commands to a robot that executes them autonomously [19].

The strategies considered in the current work can be described using these models as follows:

- Strategy 1 is an instance of direct control, in which the user directly controls the gripper in Cartesian space.
- Strategy 2 is a form of shared control, in which the user selects desired poses; the robot rejects infeasible poses and executes straight-line paths to feasible poses.
- Strategies 3 and 4 are instances of supervisory control, in which the robot autonomously executes grasps set by the user; in Strategy 4 the robot also assists by providing grasp suggestions.

3. SYSTEM OVERVIEW

The overall goal of our system was for a physical robot to perform a number of grasping tasks of common household objects in a complex environment, under the control of a human operator. The system is designed for remote operation; the operator controls the robot through a separate desktop computer, with no direct visual contact with the robot itself.

The hardware we used was the PR2 personal robot, shown in Figure 1. The PR2 has two compliant, backdriveable 7-DOF arms with parallel-jaw grippers. We used two range sensors: a widely available Microsoft Kinect^{TM} mounted on the head of the robot (providing both range and color images), and a tilting laser rangefinder mounted on the chest (used for autonomous collision avoidance). The PR2 communicated with the computer running the teleoperation interface via a commodity wireless network, as we expect that any mobile robot in real households or offices will have to be untethered in order to perform useful tasks.

We developed a "point-and-click" Graphical User Interface[1] built on `rviz`, a 3D robot visualization and interaction

[1]While higher-dimensionality input methods might provide some benefits, a major advantage of a simple cursor-driven interface is the widespread accessibility of devices that provide cursor control, including tablets and devices such as head trackers for motion-impaired users.

environment in ROS (www.ros.org/wiki/rviz). It presents the user with two main displays: on the left, a real-time feed from the Kinect camera mounted on the PR2; on the right, a rendered image of the PR2 in its current posture, along with a 3D point cloud showing a snapshot of the world as seen by the Kinect. The user can point the robot's camera by left-clicking anywhere in the camera view, changing the point of view of the live camera feed shown on the left. Since the right image is rendered, its viewpoint can be moved to any position by rotating, translating, and zooming.

For the 3D point cloud of the world shown on the right in Figure 1, we use a static snapshot instead of a continuous feed of the Kinect range data. This allows a resolution that would be difficult to stream over the wireless network; it also allows for continued 3D visualization of objects that become occluded as the arm is positioned to grasp. At any time, the user can refresh this static snapshot by right-clicking on the snapshot and selecting 'refresh'. The snapshot point cloud, as well as the various interface controls for moving the gripper pose goal, are also overlaid atop the camera feed on the left, enabling click-and-drag mouse input in both views.

4. STRATEGIES FOR GRASP EXECUTION

So far, we have described the two ends of our pipeline: a robot attempting to choose and execute a grasp that allows it to secure and lift an object from a cluttered environment, and an interface allowing a user to receive information on the robot's surroundings and provide point-and-click commands. We compared four different strategies for connecting these two. In particular, we explored the spectrum from very little to a great deal of autonomous assistance from the robot, or, conversely, from high to low operator involvement.

Strategy 1: Direct Control. In the first strategy, the user directly controls the PR2 gripper in real-time by clicking and dragging a set of rings and arrows (Fig. 2). Dragging the arrows causes linear motion along each of three orthogonal axes, and dragging the rings causes rotation about the same axes. The axes can be aligned with the gripper or the world, as seen on the right side of Figure 2.

As the user drags the rings-and-arrows control, the real gripper attempts to track it in real time. This is achieved using a J-transpose control law for the Cartesian position and orientation of the gripper. The current pose of the rings-and-arrows controls is sent as the input command to the controller at 30Hz. PD control is used to generate a desired Cartesian wrench for the gripper ($\mathbf{f}$), which is converted to joint torques (τ_p) using the transpose of the Jacobian for the end-effector pose (J):

$$\tau_p = J^T \mathbf{f} \tag{1}$$

Since the arm is redundant with respect to 6-DOF positioning of the gripper, we also provide a ring control around the shoulder. This adds nullspace joint torques (τ_n) to the controller, to bias the arm towards a desired elbow posture while attempting to maintain the current gripper pose:

$$\tau_n = \mathbf{k} * (I - J^\dagger J) * \mathbf{q_e} \tag{2}$$

where $\mathbf{q_e}$ are the joint errors (away from the desired posture), $J^\dagger$ is the Jacobian pseudo-inverse, and $\mathbf{k}$ is the vector of joint gains. The total control torques applied to the robot are $\tau = \tau_p + \tau_n$.

Note that this strategy makes very few assumptions about the task being performed; it also provides the least assistance. When using this control method for grasping, the user must essentially "drag" the robot's gripper all the way from its starting position to the desired grasp location, while avoiding collisions along the way.

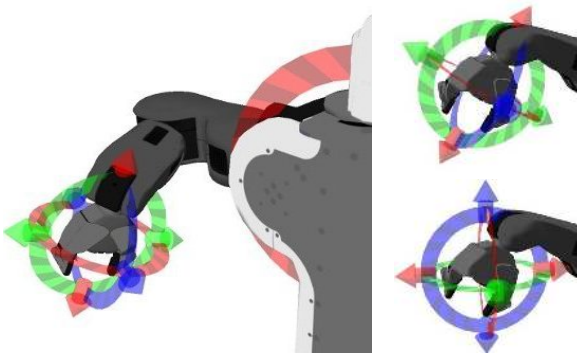


Figure 2: Strategy 1. Gripper control (6D) and shoulder ring (1D). Right images show a gripper-aligned (top) and world-aligned (bottom) control.

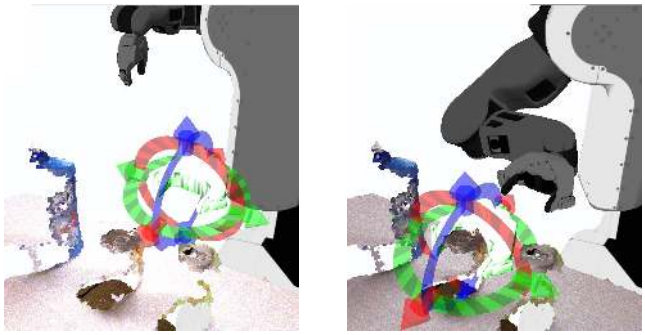


Figure 3: Strategy 2. An example of waypoint control to grab a soup can.

Strategy 2: Waypoint Following. With the previous strategy, any input from the operator is immediately tracked by the robot. This results in fast execution, but does not allow the operator to check or adjust the path of the gripper before the robot actually moves. With the second strategy, the operator uses the same type of rings-and-arrows control to move around a *virtual* gripper that specifies a new pose goal. The robot only begins to move once the user accepts the goal, which causes the gripper to move by smoothly interpolating both position and rotation to the goal pose.

When the gripper reaches the new position or detects that it is stuck, the controls reappear and can be dragged to a new waypoint. A sample sequence is illustrated in Figure 3. To reach a desired goal, this strategy uses the same controller as the previous one, but with an outer loop that essentially maintains a fixed velocity on the gripper's way to the desired goal. The shoulder control for adjusting the elbow posture is unchanged. The user can also cause the goal to instantly jump to a pose near a clicked-on point in the snapshot; the surface normal of the clicked-on point is estimated and used to initialize the direction from which the gripper approaches the point cloud (Figure 4). As an additional feedback mechanism, the virtual gripper control turns green when the desired pose is within reach, and red when it is out of reach; an out-of-reach goal cannot be accepted.

(a) (b) (c)

Figure 4: (a) In Strategies 2 and 3, the user right-clicks on a point in the cloud (marked here by a red dot) to quickly position the virtual gripper near that point, and aligned with the estimated surface normal at that point. (b) In Strategy 4, an autonomous planner shows suggested grasp poses as grey rectangle-arrow icons. The icons change color as the grasps are evaluated; green denotes a feasible grasp, while red marks a colliding or unreachable pose. (c) Clicking on a grasp icon places the virtual gripper in that pose.

Overall, this strategy attempts to take advantage of cases where specifying a new waypoint is easier than directly controlling a long, straight-line gripper motion. It also allows gripper goals to be carefully adjusted before they are executed. However, collision avoidance along the path to the waypoint is still the responsibility of the user.

Strategy 3: Grasp Execution. In the third strategy, the user only selects the final grasp pose; the robot then executes the grasp autonomously while attempting to avoid collisions along the way. The user begins by clicking a point in the 3D environment snapshot. As with Strategy 2, a virtual gripper is displayed at the clicked location and initially aligned with the local surface normal. The operator can use the rings-and-arrows control to adjust the pose as desired (Figure 4a).

Once the operator confirms the grasp location, the robot computes a pre-grasp pose, backed off from the final grasp pose by 10 cm. The robot then moves the arm into the pre-grasp pose, avoiding all collisions along the way, using the sampling-based motion planner presented in [2]. Finally, the robot moves the gripper from the pre-grasp into the grasp, avoiding collisions on the arm only. Gripper collisions are accepted in this step so that objects can be retrieved from clutter; surrounding objects may need to be pushed aside.

As the user is adjusting the virtual gripper indicating the desired grasp, the robot continuously computes whether the motion planning and collision checking components consider the grasp feasible. If a collision-free pre-grasp, and an acceptable approach from pre-grasp to grasp can be computed, the virtual gripper control turns green. If either of those computations fails, either due to potential collisions or because the desired grasp is out of reach, the virtual gripper control turns red and prevents the goal from being accepted.

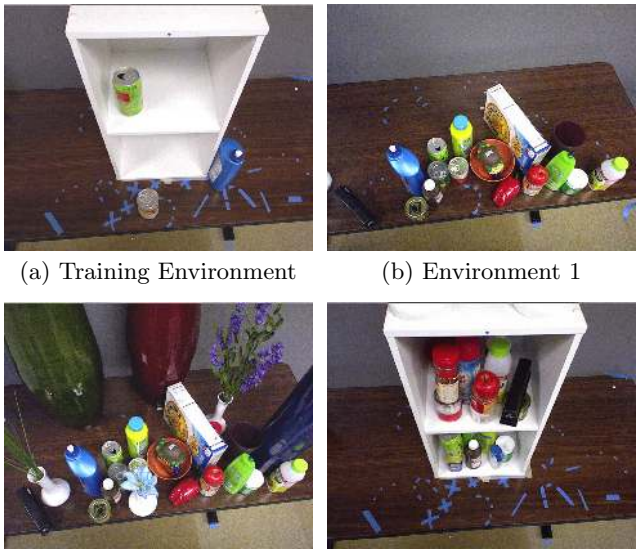
This strategy gives the autonomous components the job of planning appropriate arm joint trajectories and avoiding unwanted collisions; the operator only needs to select an appropriate final grasp pose. In doing so, a number of assumptions are made that are specific to grasping tasks (such as decomposing the motion into a pre-grasp and a grasp).

Strategy 4: Grasp Planning. In this strategy, the user has access to a set of grasp poses suggested by an autonomous grasp planner. The poses are computed near a user-selected point, and show up after a few seconds as small icons that indicate the proposed wrist poses (Fig. 4b).

The grasp pose icons show up as grey initially, which means that they are not yet checked for collisions; each pose in turn is checked in the background, and the associated icon turns red or green depending on whether the pose is deemed acceptable. Users can click on an icon of any color at any time, which causes the virtual gripper to be displayed at that grasp pose. The user can adjust this pose to their satisfaction as in the previous strategy, with the same type of red/green feedback for which poses are considered feasible by the automated components. Once the user confirms a grasp pose, execution proceeds as in the previous strategy.

When computing grasp suggestions, the robot first attempts to segment objects in the cluttered scenes by estimating surface normals for all points within a 30 cm cube around the user's clicked position, removing all points whose normals are not within 45 degrees of a given direction (up for the scenes with objects on the table; forward for the shelf), and then performing Euclidean clustering. This process separates most objects from each other, and from the table/shelf. The resulting clusters are then fed, individually, to the grasp planner presented in [9], which computes grasps along the principal axes; table scenes are limited to overhead grasps, and the shelf scene is limited to grasps from the front.

This strategy uses the most autonomy; in an ideal case, the user only has to click once around the desired objects to start the grasp planner, and once more to select one of the resulting grasps. Note that this is as much autonomy as can be provided without higher levels of semantic perception and scene understanding (the robot has no way of knowing which of the objects the user actually wants to grasp, or how to



(a) Training Environment (b) Environment 1

(c) Environment 2 (d) Environment 3

Figure 5: Environments used in this study.

distinguish objects to grasp from obstacles). However, since the autonomous components do not always behave in an ideal way, the user still has tools for adjusting the planning results, or choosing a different grasp altogether.

5. CONTROLLED EXPERIMENT

We performed a controlled experiment to assess the capabilities of the different HitL strategies to perform grasping tasks with novice robot operators, and to quantify how the various levels of autonomous assistance affected their performance. We used three environment types (Figure 5) to simulate different situations that can be encountered in a household setting, resulting in a 4 x 3 experiment (grasping strategy: 1 vs. 2 vs. 3 vs. 4, between-participants x environment type: 1 vs. 2 vs. 3, within-participants). 48 adults (24 men and 24 women) participated in this study. Each user was randomly assigned to one of the four strategies, genders were balanced across the conditions, and environment types were balanced for order within each condition. Each participant was given a $20 gift card as a token of thanks.

5.1 Participants and Procedure

Participants were recruited from local mailing lists and contacts. 3 participants were 18-20 years of age, 28 were 21-29, 13 were 30-39, 0 were 40-49, 2 were 50-59, and 2 were 60-69. These participants were not very familiar with robots (M=1.58, SE=0.15; 1= not familiar to at all, 7 = very familiar. 13 participants had no experience with playing video games; 13 had played 2D video games; and 22 had played at least one 3D video game.

Upon signing the study agreement form, each participant was shown two tutorial videos — one general and one specific to the randomly assigned grasping strategy. Next, the participant was talked through grasping the three objects in the training scene shown in Figure 5a, including how to move the robot camera, how to drop off objects, and how to refresh point clouds. Data collection then occurred over three rounds, one for each environment type. For each round of grasping, the participant had 10 minutes to grasp as many objects as possible from the given environment.

The first environment contained objects in a cluttered pile on a table, the second contained the same arrangement with the addition of six fragile-looking vases (obstacles), and the third contained a small shelf full of objects (Figure 5b-d). The full task simulated clearing the scene by grasping objects and dropping them in a container to the right side of the robot; however, because we are concerned only with the grasping of objects, and not with transporting them after grasping, we manually removed from the gripper objects that were grasped and lifted. The arm started each grasp to the side of the robot as if the previous object had been dropped off in a container; to reset the arm to that position we provided a command that simply dropped the grasped object and moved the arm back to the initial side position.

Each scene contained more objects than could be grasped in the allotted time even by expert users, so no one was able to clear all of them. Grasping stacked or nested objects only counted as one grasped object, since only one grasp was involved; the experimenter returned the additional objects to the scene. After each round of grasping, the participant was presented with a questionnaire about their experience during the task. At the end of all three rounds, the participant filled out a demographics questionnaire, and was then debriefed about the purposes of the study.

5.2 Measures

Behavioral Measures: The task performance metrics for each 10-minute round were number of (a) successful grasps, (b) major collisions, and (c) minor collisions. Because the PR2 arms are compliant and unable to exert large forces, collisions with static structures become fairly harmless. We therefore counted both *minor* and *major* collisions. *Minor* collisions were undesirable but relatively consequence-free, such as hitting the table or brushing a shelf or vase without moving it from its footprint. *Major* collisions were forceful enough to move or knock over the shelf or vases, or to potentially cause damage to either table or robot.

Self-Reported Measures: To measure the cognitive load experienced during each round, we used the NASA-TLX scale [5]. We also asked about how well a set of adjectives described the participant's user experience (easy, tedious, boring, engaging, difficult, simple, straightforward, complicated), using a 5-point Likert scale. Finally, we measured demographics, including age, gender, video gaming experience, locus of control [17], and familiarity with robots.

5.3 Data analysis

For the behavioral and cognitive load analyses, we used a mixed model analysis of covariance (ANCOVA), using strategy type as a between-participants independent variable, environment type as a within-participants variable, and video gaming experience as a covariate (0 = no experience; 1 = experience with only 2D games; 2 = experience with at least one 3D game). When significant main effects were found, we ran follow-up pairwise t-tests (with Bonferroni adjustments and video gaming experience as a covariate) to see which of the specific conditions were different from each other. If interaction effects were found with environments, we ran follow-up ANCOVAs (with Bonferroni adjustments and video gaming experience as a covariate) to see which of the results differed in each of the different environments.

For the attitudinal analyses, we used a mixed model analysis of covariance (ANCOVA), using strategy type as a between-participants independent variable, environment type as a within-participants variable, and familiarity with robots as a covariate (1 = not familiar at all; 7 = very familiar).

6. RESULTS

6.1 Behavioral Results

Number of Objects Grasped: Strategy type ($F(3, 43)=14.61$, $p< .001$, $\eta^2 = .18$) and video gaming experience ($F(1,43)=11.65$, $p<.01$, $\eta^2 = .05$) affected how many objects people successfully grasped within the 10 minute limit. Controlling for video gaming experience (set to the average of 1.19), people who used Strategies 3 and 4 successfully grasped more objects than Strategies 1 (1 vs. 3, $p>.01$; 1 vs. 4, $p<.01$) and 2 (2 vs. 3, $p>.001$; 2 vs. 4, $p<.001$). Grasped object counts were as follows: Strategy 1 ($M=$ 3.71, SE=0.42, Max=7), Strategy 2 ($M=$ 3.11, SE=0.42, Max=8), Strategy 3 ($M=$ 6.19, SE=.42, Max=11), Strategy 4 ($M=$ 6.13, SE=0.42, Max=12). People who had 3D video gaming experience generally grasped more objects (M=5.6, SE=0.3) than people who only had 2D video gaming experience (M=4.2, SE=0.4) or no video gaming experience (M=3.9, SE=0.4). See Figure 6a, which displays unmodified mean and standard error values (not controlling for video gaming experience).

There was also a significant interaction effect between strategy and environment type, $F(6,86)=3.06$, $p< .01$, $\eta^2 = .02$, which means that different strategies influenced task performance in different ways, depending upon the environmental setting. In Environment 1, people who used Strategies 3 or 4 were able to grasp more objects than people who used Strategy 1 (1 vs. 3, $p>.01$; 1 vs. 4, $p<.001$) or Strategy 2 (2 vs. 3, $p>.001$; 2 vs. 4, $p<.001$). In Environment 2, people who used Strategies 3 or 4 were again able to grasp more objects than people who used Strategies 1 (1 vs. 3, $p>.01$; 1 vs. 4, $p<.05$) or 2 (2 vs. 3, $p>.01$; 2 vs. 4, $p<.01$). In Environment 3, people who used Strategy 3 were able to grasp more objects than people who used Strategy 1 ($p<.001$), Strategy 2 ($p<.001$), or Strategy 4 ($p>.05$); people who used Strategy 4 were able to grasp more objects than people who used Strategy 2 ($p<.05$).

Number of Major Collisions: Controlling for video gaming experience, we found that the strategy type ($F(3, 44)=5.34$, $p< .01$, $\eta^2 = .10$) and environment type ($F(2, 86)=3.68$, $p< .05$, $\eta^2 = .04$) affected how many major collisions participants caused. People who used Strategies 3 ($p< .01$) and 4 ($p< .05$) had fewer major collisions than people who used Strategy 1. Video gaming experience was not found to be a significant predictor of major collisions. See Figure 6b, which includes some collisions caused by the motion planner (not the fault of the user).

There was also a significant interaction effect between strategy and environment type, $F(6,86)=3.30$, $p< .01$, $\eta^2 = .10$. There were no significant differences observed between strategies in Environments 1 or 3. In Environment 2, people had fewer major collisions when using Strategy 3 than 1 ($p< .01$) or 2 ($p< .05$) and fewer major collisions when using Strategy 4 than Strategy 1 ($p<.01$).

Number of Minor Collisions: Controlling for video gaming experience (set to the average of 1.19), we found that strategy type ($F(3, 43)=4.08$, $p< .05$, $\eta^2 = .09$) and environment type ($F(2, 86)=23.60$, $p< .001$, $\eta^2 = .25$) both affected how many minor collisions participants caused. Follow-up analyses showed that all of the environment types were significantly different from one another, $p<.01$. Furthermore, people who used Strategy 3 had fewer minor collisions than people who used Strategy 1, $p<.05$. Video gaming experience was not found to be a significant predictor of minor collisions. See Figure 6c.

6.2 Cognitive Load and Attitudinal Results

There were no significant effects for strategy type upon cognitive load. Controlling for video gaming experience, people experienced more mental demand from Environment 3 (M=4.38, SE=0.23) than from Environment 1(M=3.94, SE=0.19), $F(2, 86)=3.27$, $p< .05$, $\eta^2 = .003$. People who had 3D video gaming experience (M=3.69, SE=0.18) felt that they were more successful in accomplishing their tasks than people who had no video gaming experience (M=4.56, SE=0.24), $F(1, 43)=8.92$, $p<.01$ $\eta^2 = .02$, (1=perfect; 7=failure). They also felt less frustrated with the task (M=2.98, SE=0.26) than people who had no video gaming experience (M=3.95, SE=0.33), $F(1, 43)=6.73$, $p<.05$, $\eta^2 = .02$, (1=very low; 7=very high).

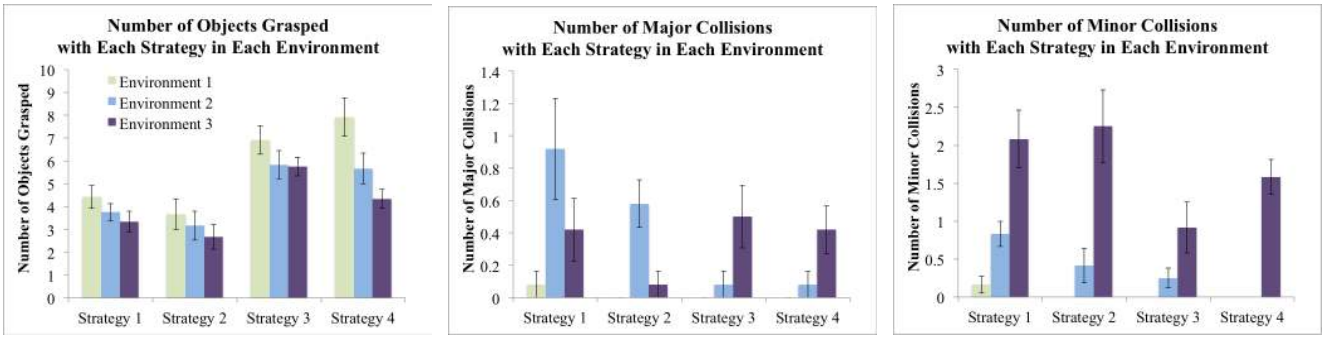
Participants also rated the following adjectives on a 5-point scale, ranging from 1 (strongly disagree) to 5 (strongly agree).

Boring: Controlling for familiarity with robots (set at the average of 3.25 on a 1-7 scale), we found that strategy type affected how bored participants felt during the task, $F(3,43)=3.07$, $p< .05$, $\eta^2 = .08$. People who used Strategy 2 felt more bored (M=2.22, SE=.0.20) than people who used Strategy 1 (M=1.56, SE=.0.20, $p<.05$), Strategy 3 (M=1.5, SE=.0.20, $p<.05$), or Strategy 4 (M=1.48, SE=.0.20, $p<.05$).

Simple: Controlling for familiarity with robots (set at 3.25), we found Environment 3 (M=2.48, SE=0.16) to be less simple than Environments 1 (M=2.63, SE=0.15) or 2 (M=2.60, SE=0.16), $F(2,86)=3.19$, $p< .05$, $\eta^2 = .01$. Robot familiarity also had a significant effect upon how simple the participants thought the task was, $F(1,43)=5.25$, $p< .05$, $\eta^2 = .04$.

6.3 Summary

Overall, the results of our controlled experiment for testing task performance using each of the four systems suggests that Strategies 3 and 4 generally produced better results than Strategies 1 and 2. In Environment 1 (cluttered scene with no fragile obstacles), people were able to grasp more objects when using Strategies 3 or 4 as opposed to Strategies 1 or 2. In Environment 2 (cluttered scene with fragile obstacles), people were able to grasp more objects when using Strategy 3 or 4 than Strategy 1 or 2. In this environment, Strategy 3 also had fewer major collisions than Strategies 1 or 2, while Strategy 4 had fewer major collisions than Strategy 1. In Environment 3 (tight shelf space), people were able to grasp more objects with Strategy 3 than with any of the other strategies. In this environment, Strategy 2 had fewer major collisions because the controller limits prevent major collisions with heavy objects; nonetheless, the differences were not statistically significant. Overall, Strategies 3 and 4 had fewer major collisions than Strategy 1, and Strategy 3 had fewer minor collisions than Strategy 1. We found no statistically significant results regarding the cognitive load



(a) Successfully grasped objects (b) Major collisions (c) Minor collisions

Figure 6: Means and standard errors for the performance metrics, not controlling for video gaming experience.

experienced between the strategies; however, environment types and video gaming experience did cause significant differences. In terms of user experience, Strategy 2 was the most boring of the four strategies.

7. DISCUSSION

One of the main goals of our experiment was to determine if, and in which ways, various level of autonomy are beneficial for a HitL robotic system driven by non-expert users performing grasping tasks. In general, the results show that the strategies with more effort unloaded onto autonomous modules performed better. The operators were able to grasp more objects, and caused fewer unwanted collisions.

7.1 Autonomy for HitL systems

Autonomous motion planning (as used in Strategies 3 and 4) proved to be highly beneficial. First, the motion planning module shielded the user from having to think about the joint configurations of the arm required to achieve various end-effector poses. This was particularly valuable as our arm has a non-anthropomorphic kinematic configuration. As noted by one user, "it was difficult noting that the PR2 wrist rotated counter intuitively from my own right hand." Second, the motion planner removed the need for the operator to think about avoiding collisions. An interesting aspect is that motion planning increased overall performance even though it occasionally exhibited failures of its own (manifested as collisions, including 7 major ones). This result suggests that autonomy does not need to be perfect in order to be helpful as a component of a HitL system.

We expected cognitive load to be lower when more autonomy was used and task performance was improved; however, this turned out not to be the case. Several users commented that waiting for the robot to execute straight-line motions to a selected pose when using Strategy 2 was boring. However, waiting for the robot to autonomously execute motion-planned grasp trajectories was not boring.

The grasp planning strategy generally performed at the same level as the grasp execution strategy, a somewhat unexpected result given its additional module in charge of suggesting grasps. A number of qualitative observations might account for this behavior, and provide suggestions for future designs. We noted that able operators often trusted the autonomous module too much. The fact that the autonomous components checked *some* (e.g. kinematic feasibility, collision-free for the arm), but not *all* (e.g. stability of object inside the gripper, gripper collisions with obstacles) characteristics of a grasp often led to confusion: operators deferred to the "checked" grasps too much rather than using their own analysis. For example, one operator noted that "because I trusted the grasp planner so much I executed without fully verifying and the grasp failed." The autonomous planner also yielded no benefit when it was not trusted enough, as operators occasionally adjusted a good suggestion into a bad grasp. One operator noted that "in a very cluttered area the automatic grasp positions weren't necessarily what I was imagining for the object [...]. For one I resorted to picking a nearby grasp location and manually moving/re-orienting to get the grasp I had in mind." Furthermore, even though the grasp planner module generally operated within about 3 seconds, even this delay was too much for some users, as they could not start adjusting the grasp pose until the grasp planner had finished. One of them noted that "the arrows took too long to show up after every move; I was frustrated." In general, our results are consistent with previous work on building trust between an autonomous system and an operator [10]. We found that it is very important for the operator to know what he or she is responsible for, and also important to be aware of what the robot does not know or is incapable of doing on its own.

7.2 Interface observations and future work

A number of observations concerned the user interface components common to all strategies. The operator's comfort with a general 3D GUI and related operations such as positioning a virtual camera proved to be very important. Those with 3D gaming experience generally performed better, though we note that a better control in future studies would be user experience with mouse-based icon manipulation, as is common in CAD and other software besides just 3D video games.

The 3D rendering component of our interface displayed a static point cloud representation of the world that could be refreshed on demand, as opposed to a continuously updating or streaming scene. This choice proved confusing for many operators, causing them to occasionally operate on out-of-date scene representations and try to grasp objects that had moved. Some offered suggestions for automatic point cloud

refreshing; one user noted that "you might as well refresh the point cloud at the end of every 'drop and return' command."

From a task perspective, we note that our system could not completely eliminate undesired collisions, regardless of the strategy used. Avoiding collisions between the gripper's protruding knuckles and the cabinet shelf proved particularly difficult, compounded by the fact that only the shelf's front edges were generally visible to the robot's sensors (and in the point cloud snapshot). Some such collisions could be avoided by improvements to the motion planning and sensing modules of our system. However, we note that grasping requires making contact at least with the grasped object, and occasionally with surrounding objects as well. This requires the ability not only to avoid collisions, but also to reason about the scene and distinguish unwanted contacts from acceptable ones, which users were often unable to do. Visual cues highlighting possible collisions would help; additional semantic labeling, either autonomous or user-aided, would be required to avoid collisions with objects not visible to the robot's sensors.

For general mobile manipulation tasks, we envision a HitL system providing access to multiple strategies such as these. Using our study environments as examples, Environment 1, with few obstacles, is better suited for grasp planning than Environments 2 or 3. Likewise, even though we allow collisions along the final grasp approach so that surrounding objects can be shoved aside during a grasp (a technique often necessary in cluttered environments), the more autonomous strategies' motion planners would prevent a user from grasping objects such as a bowl at the back of a cabinet. In future work, we hope to examine how having multiple strategies available increases the operator's efficiency when performing tasks.

8. CONCLUSIONS

In this paper, we have presented a HitL robotic system that enables novice users to remotely operate a robot performing grasping tasks in highly cluttered environments. In order to increase performance, quantified by the number of objects grasped and collisions incurred in a fixed time interval, our system uses a set of autonomous modules, such as motion or grasp planning, to assist the operator. In a controlled experiment (N=48), we evaluated the effect of these components by comparing four different grasping strategies, spanning the spectrum from very little to significant amounts of autonomous assistance for the user.

Our results show that the strategies where autonomous modules took more responsibility for parts of the task performed better than those where the operators were required to handle more by themselves. However, autonomous components must establish an appropriate level of trust with the operator in order to provide significant benefits, and communicate their limitations in an appropriate way. We believe these results can inform the design of future, more general HitL manipulation systems, which can in turn accelerate robot deployment into complex, unstructured environments.

References

[1] A. Bettini, P. Marayong, S. Lang, A. Okamura, and G. Hager. Vision-Assisted Control for Manip. Using Virtual Fixtures. *IEEE Trans. on Robotics*, 20(6), 2004.

[2] M. Ciocarlie, K. Hsiao, E. Jones, S. Chitta, R. Rusu, and I. Sucan. Towards reliable grasping and manipulation in household environments. In *ISER*, 2010.

[3] H. Das. *Kinematic Control and Visual Display of Redundant Teleoperators*. PhD thesis, MIT, 1989.

[4] W. Griffin, W. Provancher, and M. Cutkosky. Feedback Strategies for Telemanipulation with Shared Control of Object Handling Forces. *Presence: Teleoperators and Virtual Environments*, 14(6):720–731, Dec. 2005.

[5] S. Hart. Nasa-task load index (nasa-tlx); 20 years later. In *HFES*, 2006.

[6] S. Hayati and S. Venkataraman. Design and implementation of a robot control system with traded and shared control capability. In *ICRA*, 1989.

[7] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. Sensor-Based Space Robotics–ROTEX and Its Telerobotic Features. *IEEE Transactions on Robotics and Automation*, 9(5), 1993.

[8] E. Horvitz. Principles of mixed-initiative user interfaces. In *CHI*, 1999.

[9] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Jones. Contact-reactive grasping of objects with partial shape information. In *IROS*, 2010.

[10] J. Lee and K. See. Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Erg. Society*, 46(1):50–80, 2004.

[11] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, pages 30–41, 1994.

[12] M. Maybury. Intelligent user interfaces: An introduction. In *IUI*, pages 3–4, 1999.

[13] P. Michelman and P. Allen. Shared autonomy in a robot hand teleoperation system. In *IROS*, 1994.

[14] M. Oda, N. Inaba, Y. Takano, S. Nishida, M. Kayashi, and Y. Sugano. Onboard local compensation on ETS-W space robot teleoperation. In *IEEE/ASME Intl. Conf. on Advanced Intelligent Mechatronics*, 1999.

[15] M. K. O'Malley, A. Gupta, M. Gen, and Y. Li. Shared Control in Haptic Systems for Performance Enhancement and Training. *Journal of Dynamic Systems, Measurement, and Control*, 128(1), 2006.

[16] H. Pongrac, A. Peer, B. Farber, and M. Buss. Effects of varied human movement control on task performance and feeling of telepresence. In *EuroHaptics*, 2008.

[17] J. Rotter. Generalized expectancies of internal versus external control for reinforcements. *Psychological Monographs*, 80, 1966.

[18] S. Schneider and R. Cannon. Experimental object-level strategic control with cooperating manipulators. *Intl. Journal of Robotics Research*, 12(4), 1993.

[19] T. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, Cambridge, MA, 1992.

[20] E. You and K. Hauser. Assisted Teleoperation Strategies for Aggressively Controlling a Robot Arm with 2D Input. In *RSS*, 2011.