"I Want That": Human-in-the-Loop Control of a Wheelchair-Mounted Robotic Arm

Katherine M. Tsui[a], Dae-Jin Kim[b,c], Aman Behal[c], David Kontak[d], and Holly A. Yanco[a]

[a] *University of Massachusetts Lowell, Lowell, MA, USA*; [b] *Universal Robotics, Inc., Nashville, TN, USA*; [c] *University of Central Florida, Orlando, FL, USA*; [d] *Crotched Mountain Rehabilitation Center, Greenfield, NH, USA*

Wheelchair-mounted robotic arms have been commercially available for a decade. In order to operate these robotic arms, a user must have a high level of cognitive function. Our research focuses on replacing a manufacturer-provided, menu-based interface with a vision-based system while adding autonomy to reduce the cognitive load. Instead of manual task decomposition and execution, the user explicitly designates the end goal, and the system autonomously retrieves the object. In this paper, we present the complete system which can autonomously retrieve a desired object from a shelf. We also present the results of a 15-week study in which 12 participants from our target population used our system, totaling 198 trials.

Keywords: Human-robot interaction, visual servoing

1. Introduction

According the United States Census in 2000, over 21 million people reported themselves as having a physical disability, which is defined as "substantial limitation in the ability to perform basic physical activities, such as walking, climbing stairs, reaching, lifting, or carrying" (3). The majority of the people who reported a physical disability (52.7%) were in the 16-64 age category, followed closely by the over 64 age category (45.1%). A total of 6.8 million Americans not living in an institutionalized setting use mobility aids to allow for more independent and energy efficient movement within their environments (4). Of the people using mobility aids, 1.7 million people use scooters and wheelchairs (1.5 million using manual wheelchairs, 155,000 using power wheelchairs, and 142,000 using scooters) (4). As of 2007, the number of people who will use wheelchairs is predicted to increase 22% over the next 10 years (5), and we estimate the number of people who use wheelchair to be greater than 2 million.

Limitations in strength, range of motion, and dexterity in the upper extremities are issues for many people who use wheelchairs. These challenges may be exacerbated by challenges to an individual's postural control. People with spinal cord injury, traumatic brain injury, cerebral palsy, multiple sclerosis, muscular dystrophy, and other conditions may need assistance to overcome these challenges.

Email: ktsui@cs.uml.edu, djkim@universalrobotics.com, abehal@mail.ucf.edu, david.kontak@crotchedmountain.org, holly@cs.uml.edu
Parts of this paper have been presented at the 2008 Conference on Human-Robot Interaction (1) and the 2009 International Conference on Robotics and Automation (2).
D.J. Kim was affiliated with the University of Central Florida for the duration of this project. He is now at Universal Robotics, Inc.

Many activities of daily living (ADLs) involve reaching and grasping. They include tasks such as grabbing a can of soup out of the cupboard, pouring a glass of milk, and picking up an object from the floor or nearby table. Effective assistance with these tasks could greatly improve the independence and quality of life for many individuals.[1]

The Manus Assistive Robotic Manipulator (ARM) is a commercially-available, wheelchair-mounted robotic arm developed by Exact Dynamics which retails for aproximately $30,000 USD (6; 7). It is designed to assist the ADLs that require reaching and grasping and can function in unstructured environments. As purchased, the Manus ARM can be operated using a keypad, joystick, or single switch using hierarchical menus.

Römer et al. describe the process for obtaining a Manus ARM for the Netherlands (8). The potential user must meet the following criteria set forth by Exact Dynamics:

- "have very limited or non-existent arm and/or hand function, and can not independently (without the help of another aid) carry out ADL-tasks,
- "use an electric wheelchair,
- "have cognitive skills sufficient to learn how to operate and control the ARM,
- "have a strong will and determination to gain independence by using the ARM,
- "have a social environment including caregivers, friends, and/or relatives who encourage the user to become more independent by using the ARM."

Thus, the Manus ARM is largely suited to users who have limited motor dexterity and typical cognition.

Operating the Manus ARM via the menu hierarchy can be frustrating for people who have cognitive impairments in addition to their physical disabilities. They may not be able to independently perform the multi-stepped processes needed for task decomposition. They may also have difficulties with the varying levels of abstraction needed to navigate the menu hierarchy. Thus, we have investigated alternative user interfaces for the Manus ARM and have created automatic control.

The trajectory of a human arm picking up an object consists of two separate events: gross reaching motion to the intended location, followed by fine adjustment of the hand (9). We decompose object retrieval by a robotic arm into three parts: reaching for the object, grasping the object, and returning the object to the user. Our research has addressed the creation of a complete system: human-robot interaction, gross motion of the robotic arm, object recognition, fine motion of the robotic arm and gripper, grasping the object, and returning the object to the user.

The most frequent activity of daily living is object retrieval (10). Thus, our goal was to simplify the "pick-and-place" ADL. Our interface for the Manus ARM allows the user to select the desired object from a live video feed that approximates the view of the user in the wheelchair. The robotic arm then moves towards and returns the object without further input from the user.

2. Related Works

Because of the high level of cognitive ability required to operate the Manus ARM and other general purpose assistive robotic arms, several research institutions have

[1]Some individuals may be able to adapt their living environment to accommodate these limitations, but adaptations are not always possible. Moving all necessary items to within reach (when reach is extremely limited) severely limits the quantity of items available. Limits to strength and grasp may limit the effectiveness of nearly all environmental modifications.

investigated alternative interfaces and increasing the level of autonomy of robotic arms. At TNO Science & Industry and the Delft University of Technology, researchers augmented a Manus ARM with cameras, force torque sensors, and infrared distance sensors. Their alternative interface featured a "pilot mode," which was Cartesian control with respect to the task frame from a camera mounted on the workspace, facing the robot (11; 12). Looking at the video feed, the user manipulated a joystick to move a white cross hair over the object and pressed a switch to toggle the robotic arm's automatic visual servoing. Manual corrections for the robot's position could be made using their graphical user interface (GUI). Their robotic arm was capable of picking up unmarked objects using color tracking.

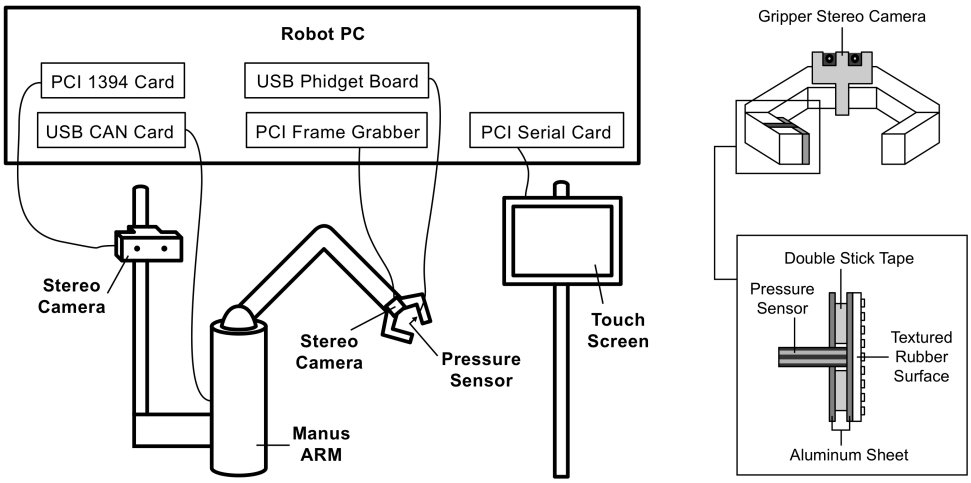
TNO's user interface with "pilot mode" begins to help reduce some of the cognitive load associated with operating robotic arms. However, the GUI presents a third person view of the workspace, which would require additional mental translations when needing to operate the robotic arm manually for correction (13).

Researchers at INRIA (Institut National de Recherche en Informatique et en Automatique) have explored a "one click" computer vision approach (14; 15; 16). Their robotic arm was equipped with two cameras. The "eye-in-hand" omnidirectional camera mounted on the robotic arm's shoulder provided an overview of the workspace and the "eye-to-hand" stereo camera offered a detailed view of the scene. The robotic arm moved toward the desired object using a visual servoing scheme along the corresponding epipolar line. Their robotic arm was capable of determining how to pick up unknown objects from a highly textured background. This system has not yet been tested with end-users.

At the Georgia Institute of Technology and the Emory School of Medicine, researchers have also investigated a single click selection approach with a full retrieval from flat surfaces (17; 18; 19). El-E is a mobile assistive robotic arm which can autonomously retrieve a variety of household objects. El-E has an omnidirectional camera on a pan-tilt unit, a high-mounted stereo camera, a Hokuyo URG laser scanner, and a color "eye-in-hand camera." Using a touch screen or laser pointer devices, a user can direct El-E to retrieve unmarked objects or open drawers and doors flagged with red towels.

El-E has been specifically designed to assist people with Amyotrophic Lateral Sclerosis (ALS, also known as Lou Gehrig's disease) and has been evaluated with eight end-users. The laser pointer interface essentially places a selection "cursor" in the real-world, which removes a layer of abstraction and decreases the user's cognitive load. However, their alternative touch screen interface may increase the user's cognitive load if the desired object is not shown on the display. Because El-E is not necessarily co-located with the user, an additional mental translation is required because the user must take the perspective of the robot when using the touch screen interface (13). In our own research, we have found that users least liked controlling a pan-tilt camera to put the object in view (1).

Our goal was to create a robotic system which could be used by people with cognitive impairments. Our interface is simply an over the shoulder video feed displayed on a touch screen. The user can indicate "I want that" by pointing to an object. With only this single selection, the robotic arm reaches towards the object, grasps it, and brings it back to the user. Further, we wanted to make the interface compatible with multiple access devices. In addition to the touch screen, we support a mouse-emulating joystick and single-switch scanning. In this paper, we show that end-users with lower levels of cognition are able to successfully use our system.



(a) Our Manus ARM has been augmented with stereo cameras (over shoulder and in gripper) and a pressure sensor within the gripper. A touch screen or mouse-emulating joystick serve as the input device.

(b) Diagram of the gripper and pressure sensor from overhead view.

Figure 1.

3. Hardware

The Manus ARM is a 6+2 degrees of freedom (DoF) wheelchair-mounted robotic arm with encoders and slip couplings on each joint. It weighs 31.5 pounds (14.3 kg) and has a maximum reach of 31.5 in (80 cm) from the shoulder (6). The gripper can open to 3.5 in (9 cm) and has clamping force of 4 pounds-force (20 N). The payload capacity at maximum extension is 3.3 pounds (1.5 kg).

The Manus ARM is programmable. The encoders' values are used for computer control. The Manus ARM communicates through controller area network (CAN) packets, sending status packets at a rate of 50Hz to a CAN receiver. It can be operated in joint mode, which moves the joints individually, or Cartesian mode, which moves the gripper of the Manus ARM linearly through the 3D *xyz* plane from the wrist joint.

We have made several augmentations to our Manus ARM to improve the user interaction and computer control (Figure 1a). We added a vision system consisting of two stereo camera systems, one mounted over the shoulder on a fixed post and one mounted on the gripper. The shoulder camera system is a color Videre stereo camera (STH-DGCS-STOC-C model), which provides the perspective of the user in the wheelchair. The live-video graphical user interface is streamed via the firewire connection from the shoulder camera's left eye, which is closest to the user's view point since our Manus ARM is a right side mounted arm. The video stream is 640 pixels $\times$ 480 pixels at 15 frames per second. The focal length is 0.12 in (3.0 mm) and the baseline is 3.54 in (9 cm). The viewing angle of each eye is 60°.

The gripper camera system provides a close-up view of the items for object recognition and autonomous grasping. The custom stereo camera uses two small PC229XP CCD Snake Cameras (Figure 1b). Each camera CCD measures 0.25 in $\times$ 0.25 in $\times$ 0.75 in (11 mm $\times$ 11mm $\times$ 18 mm). There are 6 in (25 cm) of cable between the lenses and the computational boards, which are mounted to the outside of the gripper. Each camera has 470 lines horizontally. Its viewing angle is approximately 50°, and the capture mode is NTSC with 379,392 effective pixels. The gripper stereo camera was calibrated using a Matlab camera calibration toolbox using images with 320 pixels $\times$ 240 pixels (20).

We augmented the arm by mounting a "pressure" sensor to the inside of one of the gripper's fingers. We used a CUI Inc. SF-4 1.5KGF force sensor; the sensor senses a change in resistance when force is applied. Because the sensing area is small (0.16 in $\times$ 0.16 in (4mm $\times$ 4mm)), we constructed a distributed bumper pad which is the length of the tip of the gripper's finger (Figure 1b). We removed the grip material and adhered a sheet of aluminum for support because the finger is convex. We then mounted the pressure sensor in the middle of the aluminum sheet and placed double stick tape on either side to provide a cushion against impact. We adhered another sheet of aluminum to the double stick tape and replaced the original grip material. To relieve tension on the pressure sensor's cable, we firmly tied the cable to the outside of the gripper.

We have also replaced the Manus ARM's standard access methods with a touch screen and assistive computer input device. The touch screen is a 15-inch Advantech resistive LCD touch screen. The assistive computer input device is a USB Roller II Joystick which emulates a mouse. The computer that interfaces with the Manus ARM is a 2.66GHz Intel Core2 Quad (Q9450) with 4 Gb RAM running Windows XP. The PC has a four-channel frame-grabber to accommodate the gripper stereo camera and a 1394 firewire card for the shoulder stereo camera. We access the Manus ARM's CAN bus with a GridConnect USB CAN adapter.

4. Software Architecture

To operate the robot arm, the user selects an object from the display (Step 1). Figure 2 shows how the robot arm is able to autonomously retrieve the desired object; in this case, a bottle of water. The position of the bottle is computed using stereo vision from the shoulder camera, and the robot arm moves to that position using the encoder values (Step 2). With the bottle in view, the robot arm computes a more precise target and adjusts its orientation so that the gripper is perpendicular to the object, if needed (Step 3). The robot arm compares the left gripper camera image with the fifty template images in the database and chooses the best match (Step 4). Using the chosen template, the robot arm moves to align the feature points (Step 5). Once aligned with the template image, the robot arm moves forward and closes its gripper (Step 6). The robot arm then returns the bottle of water to the user (Step 7).

Our software architecture consists of multi-threaded modules in order to ensure timely response. The system data flow is shown in Figure 3. The graphical user interface displays the shoulder camera view, which is approximately the view of the user in the wheelchair. The object selection module streams the live image feed from the camera and also computes the position of the object. The coordinator module continuously streams the gripper camera view and is also responsible for inter-thread communication. The visual tracking module recognizes the object from a template database and tracks the features while the Manus ARM picks up the object. The robot control module controls the trajectory of the Manus ARM, grasps the object, and returns it to the user.

4.1. User Interface

When designing an intuitive interface for the Manus ARM, we leveraged two well-established access methods: a joystick and touch screen. Many people in our target population are already able to drive a powered wheelchair with a joystick or use a mouse-emulating joystick to control a computer. Additionally, these users may have

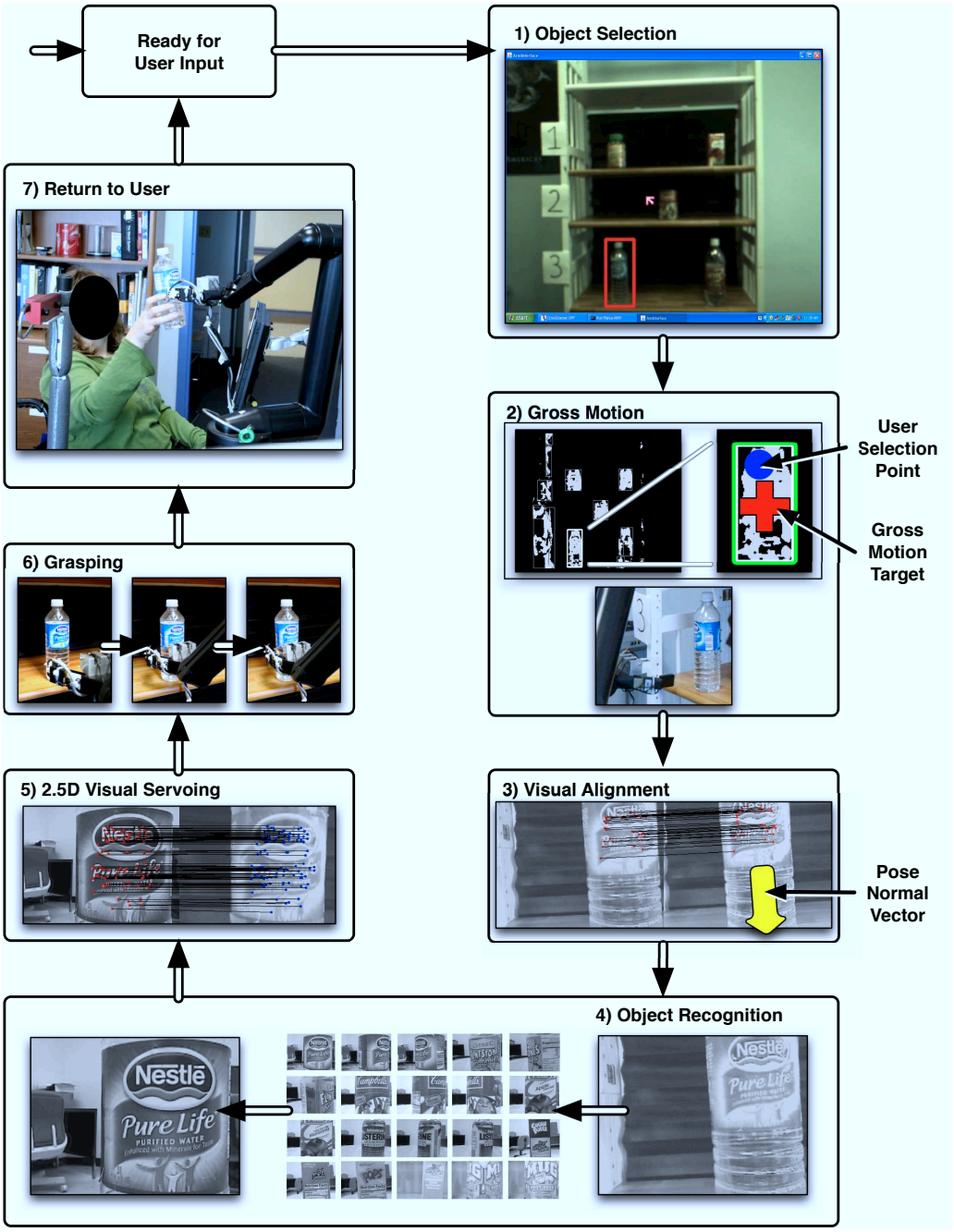


Figure 2.: The user provides input via touch screen or joystick. The input is used in the vision processing to position the robotic arm. (Best viewed in color.)

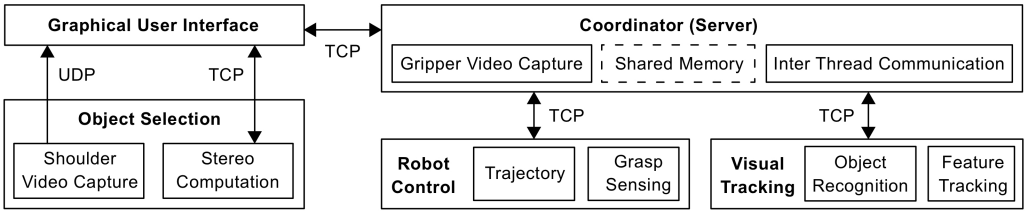


Figure 3.: Data flow of our direct-selection, human-in-the-loop visual control system for the Manus ARM.

Algorithm 1 Gross Motion

```
1: procedure GrossMotion(point of interest, shoulder stereo image pair)
2:     Calculate disparity map ← left and right images
3:     Get disparity value ← point of interest
4:     Generate blobs ← disparity map, disparity value
5:     Find largest blob ← blobs, region of interest
6:     Get associated world coordinates ← blob
7:     Translate to world coordinates
8:     return (X, Y, Z in world coordinates relative to F_shoulder camera)
9: end procedure
```

some experience with touch screens, which are commonly used as communication devices.

The interface shows the shoulder camera view, which is similar to the user's view. The shoulder camera's left eye is captured using the Small Vision System (SVS) and continuously streamed to the interface via the UDP network protocol (21). The interface displays the fullscreen image. We provide a natural interaction in that the user indicates "I want that" by pointing to an object on the touch screen.

We simplified the input device to either a single press on a touch screen or the use of a mouse-emulating joystick. For the interface using the joystick, we selected a large (64×64 pixels) cursor for the interface because of the range of each person's vision ability within our target population. The cursor is enhanced with a white outline to provide visibility against a dark background. The cursor speed can be set from the Windows Accessibility panel.

A dwell configuration is available for users who are not easily able to move between a joystick and a button. According to the Open Source Assistive Technology Software (OATS) Project Consortium, "dwelling is resting the mouse over one area of the screen for a specified time" (22). The system interprets a mouse click when the cursor remains stationary for a period greater than the set dwell length. The dwell length should be a long enough interval to prevent accidental clicks.

Our feedback to the user is multimodal. When an object is selected and the system is able to correctly identify it, a bold, red rectangle highlight is drawn around the object (Figure 2) and a "ding" sounds. When the system is unable to segment the object, a "Please try again" prompt sounds. Also, the robot arm provides audio prompts at the start of each algorithmic phase to keep the user aware of the current step in the process (e.g., "I am trying to identify the object you want" after gross motion, "Ok, I have figured out where to put my fingers" after fine motion). Additional prompts, such "I am still thinking" and "Maybe I should try over here," are spoken at every 5, 10, 20, 30, or 60 seconds depending on the user's preference to keep the user aware that the system is still functioning.

4.2. Gross Motion

Given the user's selected point of interest (POI, shown as a blue dot in Step 2 of Figure 2), we must move the gripper towards this target position in 3D (Algorithm 1). First, the disparity between the left and right stereo images is calculated using SVS. We filter the disparity image with a mean shift segmentation filter to further group similar depth pixels. We then calculate the associated depth layer of the point selected within 10 pixels. Given this depth layer, we find similar "blobs" (i.e., continuous regions of pixels) on the disparity map using OpenCV and the

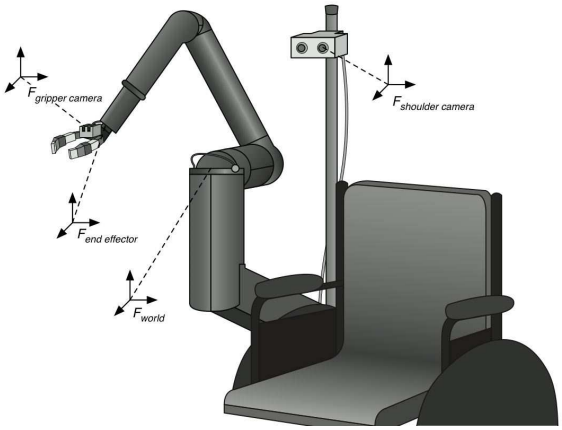


Figure 4.: Four coordinate frames are used to direct the Manus ARM's end effector to retrieve an object. The center of our coordinate system is located at the top of the robot's shoulder in the center (F_{world}).

cvBlobsLib libraries[2] (23; 24). We filter the blob noise and constrain blobs to be a size which the gripper can grasp. We select the largest blob for which the center of bounding box is within 55 pixels of the selection point (Figure 2).

Using the center of the bounding box of the selected blob (shown as a red cross in Step 2 of Figure 2), we then calculate the 3D coordinates of that pixel location. If there is not enough texture, we compute the 3D coordinates of the nearby pixels in an outward, clockwise spiral. The $<X,\ Y,\ Z>$ returned are values in world coordinates (in millimeters) with respect to the center of the left-eye ($F_{shoulder\,camera}$ in Figure 4), X is to the right of the center of the lens, positive Y is to downward of the center of the lens, and positive Z is outwards from the lens.

As shown in Figure 4, the shoulder camera is fixed to the Manus ARM. We have calculated offsets from $F_{shoulder\,camera}$ to determine the relative target position to the ARM's shoulder (F_{world}). These F_{world} coordinates are the position of the desired object. In the gross motion, we want to direct the Manus ARM's end effector towards the selected object without knocking it over or pushing it away; we subtract 12 in (307 mm) from the depth. The Manus ARM's end effector then moves to this target position.

4.3. 3D Geometry Estimation and Visual Alignment

Now that the object is in the field of view of the gripper camera, we want to find a more exact 3D target using an analysis of image features. First, we compute a 3D point close to the object. Then we compute a normal to a locally approximated plane around the POI.

Algorithm 2 details the computation of the local 3D geometry. We use the Scale Invariant Feature Tracking (SIFT) descriptor to estimate the 3D position for the target object from the gripper stereo camera (25). We utilize the SIFT descriptor because of its invariance with scale, rotation, and illumination conditions.

Given an initial set of SIFT descriptors, we match the descriptors between the left image and the right image using epipolar constraints (Figure 5). To make the match, we minimize the error between the measured point from one side to the projected point on the other side using a least-squares linear optimization.

[2]OpenCV is an open source computer vision library (23). cvBlobsLib is a library in which similar regions of an image are grouped together (known as "blobs") (24).

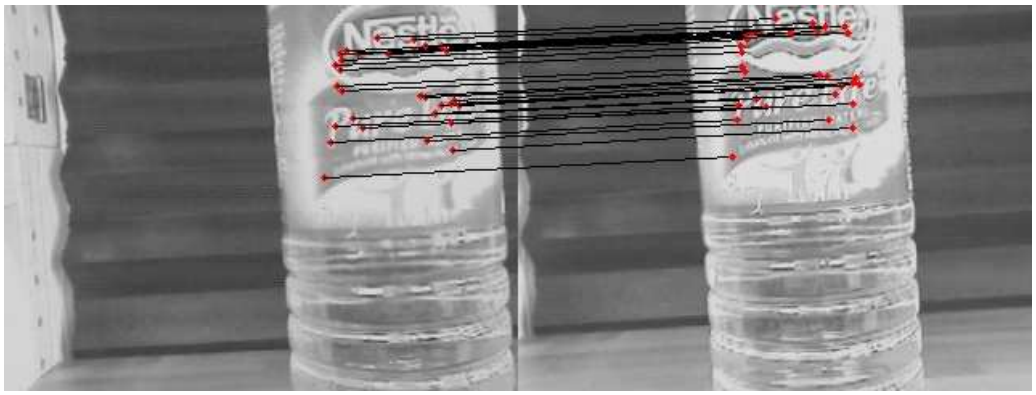


Figure 5.: 3D geometry estimation: initial SIFT descriptors matched between the left and right images of the gripper stereo camera at a distance of twelve inches.

Algorithm 2 Estimating Local 3D Geometry

1: **procedure** COMPUTE3D(gripper stereo image pair)
2: SIFT descriptors ← left and right images
3: 3D point cloud ← matched pairs of SIFT descriptors
4: Eliminate outliers ← 3D point cloud
5: Calculate best normal vector ← 3D point cloud inliers
6: **return** (3D point cloud, best normal vector)
7: **end procedure**

Algorithm 3 Visual Alignment

1: **procedure** VISUALALIGNMENT(3D point cloud, best normal vector)
2: Target 3D location $\mathbf{x}^* \leftarrow$ 3D point cloud
3: Target 3D orientation $\theta^* \leftarrow$ best normal vector
4: **repeat**
5: Translation error $\leftarrow \mathbf{x}^* - \mathbf{x}^c$ ▷ $\mathbf{x}^c$ *is current location*
6: Rotation error $\leftarrow \theta^* - \theta^c$ ▷ θ^c *is current orientation*
7: Generate 6D translation/rotation motion
8: **until** Gripper is located in the target pose
9: **end procedure**

We then apply constraints to the initial 3D point cloud to further eliminate outliers. First, we consider the depth ratio of each descriptor match. We also consider the size of the objects that the Manus ARM can grasp given the size of the gripper's maximal opening. We constrain the points to be within a six-inch cube around the POI. Lastly, we examine the distribution of the remaining points.

With this refined 3D point cloud, we compute the normal vector for an approximate local plane of the POI. We define a set of N prototype normal vectors given the possible set of poses that an object is likely to have and the camera's field of view. We have implemented four normal vectors where three vectors are for upright objects, and one is for objects that lay down; Step 3 of Figure 2 shows an example upright pose normal vector as a yellow arrow. For each point in the 3D point cloud, we compute a rank (from 1 to N) for each plane, indicating the relative likelihood of the given point being an inlier of these planes. To compute the rank, we use the estimate of the 3D point closest to the POI and a distance metric. The ranks are tallied for all points to obtain the best representative normal.

The resultant 3D geometry information is used to refine the target object position. The chosen normal is used to compute the gripper's yaw and pitch angles.

Algorithm 4 Two-Phase Object Identification

```
1: procedure ObjectID(gripper stereo image pair)
2:     SIFT descriptors ← left and right images
3:     Eliminate outliers ← SIFT descriptors
4:
5:     Phase I:                                   ▷ Object image retrieval
6:     Top 5 list of templates ← SRVT query with inliers
7:     for all top 5 list do
8:         Number of matched ← RANSAC with left image
9:     end for
10:    Candidate object image ← top 5 list
11:
12:    Phase II:                                  ▷ Object-view refinement
13:    Retrieve multi-view images of candidate object image
14:    for all multi-view images do
15:        Confidence level ← RANSAC with left image
16:    end for
17:    Final view of object image ← template database
18:    return final view image
19: end procedure
```

To move to this refined target position with optimized gripper position, we generate translational and/or rotational velocity command using a proportional control algorithm (Algorithm 3).[3]

4.4. Object Recognition

With the gripper positioned normal to the object and in close range (within 12 in (305 mm)), our next step is to recognize the object. Again, we use SIFT descriptors for their robustness. Our object recognition algorithm has two parts: object identification and object view selection (Algorithm 4). To recognize the object, we search our template image database comparing feature points.

Our system requires at least one image template of each object, and there may be multiple views of each object. In addition, a person may utilize a large number of objects in his or her ADLs. As a result, our database may become extremely large making it computationally intractable to retrieve the correct template through a sequential scan. Instead, we use a decision tree approach known as Scalable Recognition by Vocabulary Tree (SRVT) (27). SRVT consists of a multi-level decision tree and visual keywords (i.e., sets of feature points) as leaf nodes. Each leaf node can be shared between multiple template images or solely allocated to a specific template image.

Figure 6 shows the template database used in our system. It is easily extendible and scalable to deal with many different natural scenes. A user can add new objects to the database using an "add object template" procedure. For each object, the user can provide multiple views of the object on a variety of backgrounds. The SRVT implementation has been shown to take 25 ms to match an image in a database of 50,000 images (27), therefore we believe that this approach will scale well.

Our vocabulary tree was constructed using more than 40,000 frames from a set of movie clips. Keypoints representing the current frame were used to query the most

[3]Detailed mathematical descriptions of Algorithms 2 and 3 can be found in (2) and (26).

Figure 6.: Template database of 26 unique objects with several having multiple views. Objects are identified using a decision tree that compares feature points.

similar template image from the ADL template database. We observed inconsistent results and poor discrimination because of keypoints generated from features in the background. We improved the performance of the SRVT algorithm by segmenting the object from the background using stereo information and statistics.

We use a two-phase object identification approach (Algorithm 4). We match the object with an existing template. Then, we retrieve the top five template matches based on the keypoints. We compare the current image with each of these templates using RANSAC (28). This comparison yields only one possible object image. It is possible that the object is not recognized at all because either the object does not exist in the database, or the current view is a partial view (e.g., a can of soda viewed from the side whose template was taken from the front). Figure 7 shows a

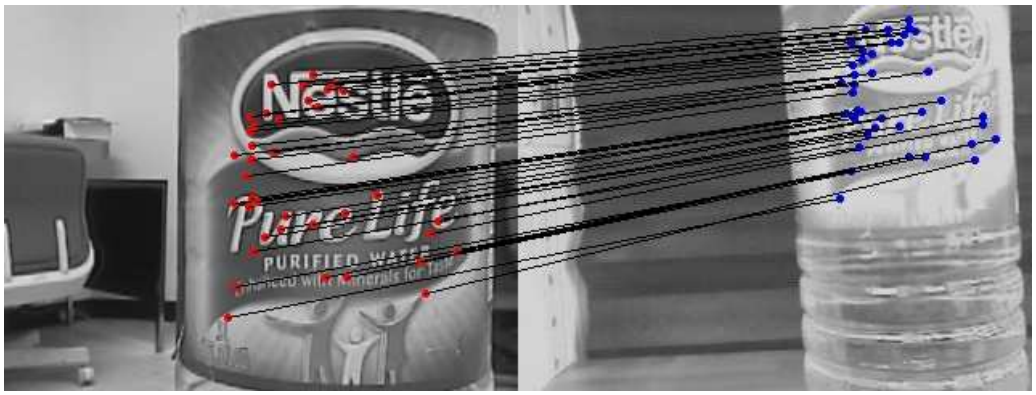


Figure 7.: Object recognition: a successful match between template (left) to current gripper camera view (right) after gross motion and visual alignment.

Figure 8.: Successful fine motion alignment between template (left) and left gripper camera view (right). (Best viewed in color.)

successful template match.

After identification, we must choose the best view of the object if there are multiple views in the template database. This refinement provides the most desirable template for grasping. If a meaningful template is not found, we use a Principle Component Analysis (PCA) based analysis to reorient the gripper (29).

4.5. Fine Motion

With the object identified and the optimal view chosen, we now compute the fine motion necessary to position the gripper around the object. We use visual servoing to guarantee precision and replace SIFT with a fast keypoint feature detector, *ferns*, to guarantee speed (30). The fine motion control is composed of a two-phase control scheme (Algorithm 5). Figure 8 shows a successful alignment with the object.

We use a 2.5D (or hybrid) visual servoing scheme to generate translation and rotation motion to align the current gripper camera view with the pertinent template database image (31). We compute the Euclidean homography relation between the current image and the template using matched pairs of local descriptors and the intrinsic camera calibration matrix of the gripper stereo camera. The computed homography is decomposed into two feasible solutions for the rotational and/or translational motions to position the gripper around the object. Only one of these solutions is physically correct, so we choose the correct solution using the third view from an auxiliary stereo frame and the extrinsic calibration parameters of the gripper stereo camera (32).

Algorithm 5 Two-Phase Fine Motion Control algorithm

```
1: procedure FineMotion(current image)
2:     local descriptors using ferns ← current image
3:
4:     Phase I:                                          ▷ x − y plane centering
5:     repeat
6:         an (trackable) anchor point m^c ← local descriptors
7:         translation error ← m^c − m^o
8:                                                  ▷ m^o is an image center point
9:         generate 2D translation motion
10:    until translation error is negligible
11:
12:    Phase II:                                                  ▷ 6D alignment
13:    repeat
14:        an anchor point m^c ← local descriptors
15:        translation error ← m^c − m^*
16:                                  ▷ m^* is a correspondent point on the template
17:        target 3D orientation θ^* ← local descriptors
18:        rotation error ← θ^c − θ^*                          ▷ θ^c is current orientation
19:        if m^c enters critical zone then
20:            generate 2D translation motion
21:        else
22:            generate 6D translation/rotation motion
23:        end if
24:    until translation/rotation errors are negligible
25: end procedure
```

For stable operation of fine motion, it is imperative to keep all of the features inside the gripper camera's field of view. We use a two-phase algorithm when approaching the object (Algorithm 5). In Phase 1, we center the gripper camera on the target object using lateral motion. Centering on the object provides a large number of local descriptors which are necessary for a high-confidence homography solution. In Phase 2, with the object centered on the gripper, we generate rotational and/or translational velocity commands in real-time using homography analysis. In order to generate trajectories that converge to the desired pose, the control alternates between Phase 1 and Phase 2 as needed. That is, the 6-DOF motion generated in Phase 2 may cause the gripper to drift off center and enter a critical zone, defined as strips 1/8 image width and height along the image borders. Phase 1 will realign the lateral motion before allowing further progress on the approach.

To generate translational motion, we choose one of the local descriptors from the current image as an anchor point to visually track. Specifically, we select a median descriptor among the inliers to provide stable control performance over time. The rotational motion is generated by parameterizing the retrieved rotation transform matrix into setpoint information. We begin approaching the object through the optical axis of the gripper camera when signaled by the ratio of the depth information of both the current pose and the final pose (33).

4.6. Object Grabbing using Force Profile

With the gripper positioned in front of the object, we can now grasp the object. The grasp must be firm enough that the object does not slip out when we bring it back to the user, but not so firm that it crushes the object. We use a force sensing

Algorithm 6 Force-Sensitive Object Grabbing

```
1: procedure ObjectGrab(force profile)
2:     start closing action of the gripper
3:     repeat
4:         n-QueueList ← force signals
5:         detect plateau ← n-QueueList
6:     until plateau is detected
7:     stop closing action
8: end procedure
```

resistor mounted inside the finger to detect the grabbing force, or pressure, during the gripper's closing motion (Figure 1b). We read the pressure values as the gripper begins to close and keep a history of these values. When the difference between the last n values is small, a plateau has been detected and the gripper stops closing (Algorithm 6). In our system, we check the last five values for the plateau.

We use the pressure sensor instead of an absolute threshold for two reasons. First, the Manus ARM needs to pick up a variety of objects. Second, the same object may have different states. For example, an empty bottle and a filled bottle need different amounts of force applied to be appropriately grabbed by the gripper. We have empirically found that this algorithm works for a wide range of ADL objects including water bottles, cereal boxes, and beverage cans.

5. Experiment

We conducted an end-user testing with the system presented in this paper at the Crotched Mountain Rehabilitation Center from July through October 2009. Twelve people participated in four sessions each, in which each session had at least four trials. The experiment had a total of 198 trials. Our goal was to explore the level of cognition required to use the system.

5.1. Participants

The assistive technology director of Crotched Mountain played a central role in recruitment of participants. Given all of the students and the residents who use wheelchairs, the director evaluated their overall physical dexterity, cognitive ability, visual ability, and access method. The inclusion criteria for this experiment was 1) the person used a wheelchair, 2) the person could attend one practice session and all four sessions, and 3) the person could select an object from the user interface with his/her access method. Candidate participants were invited to participate in the experiment; fourteen participants consented. Twelve of the fourteen participants met the inclusion criteria.[4] The remaining two people were unable to participate in this study because one participant was unable to see the objects on the shelf or on the screen, and the other had not yet recovered from surgery.

5.1.1. Ability Description

For this study, we define "high" cognitive ability as a person who has full literacy, ability to function independently within typical social or environmental situations, and the ability to successfully complete complex tasks with three or more steps. We define "medium-high" cognitive ability as a person who has moderate literacy,

[4]Participant 4 (P4) and Participant 8 (P8) participated in a previous end-user evaluation in 2007 (1).

Table 1.: Participant profiles

	Age	Sex	Diagnosis	Cognition	Vision	Behavior	Communication	Wheelchair	Access
P1	60	M	Traumatic Brain Injury	High	Corrected with glasses	No significant challenges	No significant challenges	Manual	Touch screen
P2	46	M	Traumatic Brain Injury	High	Within normal limits	No significant challenges	No significant challenges	Manual	Touch screen
P3	17	M	Cerebral Palsy	High	Corrected with glasses	No significant challenges	No significant challenges	Power with joystick access	Touch screen
P4	20	F	Cerebral Palsy	High	Corrected with glasses	No significant challenges	No significant challenges	Power with joystick access (limited dexterity in driving hand)	Touch screen
P5	20	M	Cerebral Palsy Infantyle, Mental Retardation, Seizure Disorder, Epilepsy	Medium-high	Corrected with glasses; does not tolerate wearing	No significant challenges	Non-verbal; can communicate with AAC device (very limited) and facial expressions	Power with multiple switch access	Single switch scanning
P6	60	M	Traumatic Brain Injury	Medium-high	Corrected with glasses	No significant challenges	No significant challenges	Manual	Touch screen
P7	51	M	Traumatic Brain Injury	Medium-high	Corrected with glasses; left visual field neglect	No significant challenges	No significant challenges	Manual with caregiver	Touch screen with key guard
P8	19	F	Spina Bifida	Medium-high	Within normal limits	Low frustration tolerance; needs encouragement	No significant challenges	Manual	Touch screen
P9	17	F	Cerebral Palsy	Medium	Within normal limits	Very low frustration tolerance; needs encouragement	Non-verbal; can communicate well with AAC device and thumbs up/down	Manual	Touch screen
P10	21	M	Traumatic Brain Injury, Spastic Quadriplegia	Medium	Corrected with glasses; does not tolerate wearing	No significant challenges	Non-verbal; can communicate with AAC device and thumbs up/down	Power with joystick access	Touch screen
P11	17	F	Cerebral Palsy	Medium	Within normal limits	Can be excited which causes anxiety	Below age level	Manual	Touch screen
P12	19	M	Cerebral Palsy	Medium-low	Within normal limits	Can be excited which causes giddiness	Repeats prompts verbatim	Power with joystick access (limited dexterity due to high tone in driving arm)	Touch screen with key guard and head pointer

is mostly independent in typical social and environmental situations, and requires assistance on some complex tasks. We define "medium" cognitive ability as a person who has some literacy skills, requires moderate prompting (50%) to function in typical situation, is able to successfully complete simple tasks with one or two steps, and is unable to successfully complete complex tasks. We define "medium-low" cognitive ability as a person who can identify letter and number symbols, is able to read a few words, requires significant prompting to complete ADLs, and finds simple tasks with one or two steps challenging. We define "low" cognitive ability as a person who has no literacy, requires maximum support in completing ADLs, and is unable to follow simple instructions with consistent accuracy.

All participants in this study either had a traumatic brain injury or a developmental disability. These conditions are marked by challenges that range across all functional areas. Cognition, motor skills, sensory skills, behaviors, and psychological status can all be involved to a greater or lesser degree. All of these components overlap to such a degree as to render meaningless any attempt to quantify disability status with any precision.[5] For this reason, we provide a description of participants' characteristics as a measure of relative skills (shown in Table 1).

The twelve participants had varying levels of cognition: four participants were rated as having high cognition, four with medium-high, three with medium, and one with medium-low. There were four women and eight men. The participants' ages ranged from seventeen to sixty. Seven of the participants were students of the Crotched Mountain School; five were diagnosed with cerebral palsy, one with spina bifida, and one with traumatic brain injury/spastic quadriplegia. Four of the remaining participants were patients of the Crotched Mountain Rehabilitation Brain Injury Center; all were diagnosed with traumatic brain injury. The last participant was a short term patient of the hospital who was recovering from surgery. Seven of the participants used a manual wheelchair; six of the seven were able to transport themselves, and one required a personal assistant for transportation. The remaining five participants used power wheelchairs; four of the five used joystick access, and one used a switch array.

5.1.2. Access Methods

Each participant's first session was a practice session in which we determined the most appropriate access method for using the robot. Nine participants were able to use the touch screen. Three participants required alternative access methods.

Two participants required the use of a key guard, which is a rigid plastic cover with holes directly over the keys or buttons for devices such as keyboards and touch screens. For the purposes of this experiment, we created a key guard with five selection points (i.e., top left, top right, center, bottom left, and bottom right). In general, it would be possible to use a key guard with a grid of selection squares over the touch screen for selecting an object in an unstructured environment.

Key guards offer two advantages. First, the holes direct the user's finger to the appropriate spot for activation, which overcomes the common problem of striking between the keys and activating two keys at once. Second, the rigid plastic surface provides a support surface that can be used to stabilize the user's hand. This stabilization can be useful for people who have tremors or inaccurate movements of their hand due to shoulder instability.

Participant 7 (P7) was able to isolate a finger for activation; however, accurate movements of his shoulder for directing his finger isolation were challenging. It was

[5]People with traumatic brain injury may be assigned a Ranchos Los Amigos Scale (34) score for progress in their rehabilitation. Our participants were either level 7 or 8. However, it was inappropriate to assign Ranchos scores for the participants whose diagnoses were not brain injury.

Figure 9.: Experimental setup with single switch scanning. P5 used Cross Scanner (35) single switch scanning software and activated the scanning with a head switch.

less physical effort for P7 to use a key guard and his accuracy of the selection with the key guard was better than without it. In several instances, he rested his hand on the key guard during selection.

Participant 12 (P12) initially used the mouse-emulating joystick. However, due to his emerging joystick skills, it was easier for him to use a head pointer (his previous computer access method) to select on the touch screen with a key guard. P12's head pointer was a metal rod approximately 12 in (305 mm) in length with a rubber cap on the end. The metal rod was attached to a base ball cap under the center of the cap's brim. The cap was secured in place using a chin strap.

Participant 5 (P5) used a single switch to access his AAC (alternative augmentative communication) device; he used Cross Scanner (35) to operate the robot (Figure 9). Cross Scanner is a commercially available supplemental single switch scanning software. The first switch click starts a top to bottom vertical scan. When the line is in the desired vertical location, a second click starts a left to right horizontal scan. When the cursor is over the desired horizontal location, the third click selects.

5.2. Protocol

The protocol used for this experiment was based on that of our 2007 study (1).[6] The task in this experiment was to instruct the robot arm to retrieve object from a shelf (Figure 9). The experimenter would show the participant a photo of the desired object, and ask the participant to identify it on the shelf first. The experimenter

[6]When designing our 2007 study protocol, our clinicians determined that the majority of their clients were unlikely to be able to use the manufacturer's interface without experiencing frustration.

would ask questions about the object to determine in the participant could see the object (e.g., "Is the object on the top, middle, or bottom shelf?"). When the participant had identified the object on the shelf, the experimenter instructed the participant to select the object on the user interface using his or her access method.[7] After confirmation by the experimenter, the robot would retrieve the desired object from the shelf.

We selected this matching task because it allowed the experimenter to change the difficulty of the task. Prior to the experiment, we chose four layouts in which to configure five, six, eight, and nine objects. With a simpler layout, it is easier to determine where the desired object is. Also, the size of the object and contrast of the object to the background were factors in adjusting the difficult of selecting the object. The mouthwash bottle was the smallest object in our set and had the least area displayed on the screen in which to select. Lastly, some objects were more motivating to participants than others. For example, the can of soup featured DreamWorks Animiation's Shrek and Donkey on the label which was particularly motivating to Participant 11 (P11).

Most of the participants came for a practice session and four additional sessions.[8] At the start of each session, the experimenter administered a mood rating scale. The experimenter then asked the participant to retrieve four objects. At the end of each session, the experimenter administered the mood rating scale again and a post-session questionnaire. The duration for each session averaged forty-five minutes.

Data was collected from manual logs, pre- and post-session mood rating scales, post-session questionnaires, and computer generated log files. Each session was video recorded. The mood rating scale was based on the Profile of Mood States (36). We administered a shortened version of the Psychological Impact of Assistive Devices Scale (37) after the practice, first session, and last session; the participants reported how they perceived the robot with respect to sense of control, usefulness, frustration, ability to participate, independence, confusion, sense of power, happiness, and eagerness to try new things. In the post-session questionnaire, we asked the participants what they liked most and least about using the robot, what would they change about the robot, and what would make it easier to use the robot. After the final session, we asked the participants how they liked the look of the robot, how often they would use the robot if it were mounted to their wheelchair, and to describe how they would use it.

5.3. Results and Discussion

There were 198 trials. For each run, we logged the time for user selection, gross motion, visual alignment, object identification, fine motion, grasping, and return of the object to the participant. User selection time was divided into perception time (i.e., the participant acknowledged where the object was on the shelf and on the display), and motor time (i.e., the participant selected the object on the display). For Participants 5, 7, and 12, we refined the perception time by subtracting the time in which the experimenter was speaking or setting the participants' access devices (calculated manually from the videos). From the gripper camera images logged after the gross motion and fine motion, we calculated the number of feature points in each image and template, respectively, and computed the matching feature points between the images.

---

[7]Because Participants 5, 7, and 12 required the display to be nearly directly in front of their faces in order to correctly select objects which obscured the shelf, an assistant moved the display to the side until selection time.

[8]Participant 6 attended three sessions while Participant 12 attended five sessions.

5.3.1. System Performance Analysis

We found that our system was able to successfully retrieve an object from the bookshelf in 129 of the 198 trials (65% overall). For the 129 successful trials, the average length from user perception of the object to returning the object to the user was 164.72 s (SD=61.71). The average user selection time was 23.52 s (SD=27.22) in which the average perception time was 6.26 s (SD=10.39) and motor 6.11 (SD=10.40). The average algorithm running time (i.e., gross motion to object return) was 112.2 s (SD=14.5) in which the average gross motion time was 10.01 s (SD=17.12), visual alignment 1.79 s (SD=0.27), object recognition 13.59 s (SD=5.95), fine motion 37.23 s (SD=10.93), and grasping and return 36.51 (SD=2.42). The average number of matched feature points between the left and right gripper camera images after gross motion was 64.74 points (SD=58.62).

We encountered 69 failures. In 13 of the 69 failures, one or both of the gripper cameras became disconnected due to the external sensor cables moving and pulling during the robot arm's operation. These hardware failures were unavoidable on a moving arm. A commercial version of this system would have the sensor cables within the robot arm itself. Discounting these, we had 56 algorithmic failures.

Inaccurate gross motion movement accounted for 34% of the failures (19 of 56). One of the nineteen failures was due to a protocol failure in which the gross motion was triggered before the user selection was complete. We compute the 3D position of the desired object, and the robot arm moves in Cartesian coordinates to this location. The robot arm is cable driven, and the resulting position of the gross motion may not be exactly the correct physical location. The further the robot arm had to move, the greater the error was accumulated in the encoder positions. Nine of the remaining 18 gross motion failures (50%) occurred when the robot arm moved to the top shelf; the robot arm positioned the gripper too low and the objects were not sufficiently within the gripper camera's view. Interestingly in one of these runs where the desired object was on the top shelf, the robot arm instead identified the object on the shelf below and "successfully" returned the wrong object to the participant. Similarly, in 7 runs (39%), the desired object was located on the left-hand side of the shelf and the robot arm positioned the gripper too far to the left of the object.

Three errors (5%) occurred during visual alignment and five (9%) during object recognition. In 13 of the 56 failures (23%), the robot arm timed out during the fine motion algorithm. When the robot arm has exceeded 90 seconds attempting to align to the template, it returns to the lowered home position for another selection. The desired object in 6 of these 13 runs was the root beer. It should be noted that the template for the root beer was generated with a lower contrast camera than the other objects that the robot arm can recognize (Figure 6).

In 7 runs (13%), the robot arm did not successfully grasp the object because it pushed the object out of its grasp (3 of 7), pinched the object out of its grasp (1 of 7), did not close the gripper (1 of 7), reached its maximum extension (1 of 7), or the system froze (1 of 7). In 2 runs (4%), the robot arm dropped the object while returning to the user because the objects slipped out of a weak grasp. In 6 runs (11%), the robot arm froze while returning the object to the user. There was 1 unknown failure (2%).

5.3.2. System Usability

All participants were able to use the robot arm system and performed a total of 198 selections with their various access methods. The participants selected the correct object in 91.4% of the trials (181 of 198). Tables 2 and 3 show the average perception and motor times for user selection of all 198 trials categorized by

Table 2.: User Selection: Perception Times for 198 Trials

Cognition	**No. of Participants**	**No. of Samples**	$\bar{x}$ (s)	SD (s)
High	4	67	2.94	1.40
Medium-high	4	62	7.48	7.88
Medium	3	49	7.10	9.83
Medium-low	1	20	19.76	21.37

Table 3.: User Selection: Motor Times for 198 Trials

Cognition	**No. of Participants**	**Access Method**	**No. of Samples**	$\bar{x}$ (s)	SD (s)
High	4	Touch screen	67	2.70	2.72
Medium-high	2	Touch screen	29	5.00	3.97
Medium-high	2	Touch screen with key guard; single switch scanning	33	57.69	103.49
Medium	3	Touch Screen	49	3.82	3.97
Medium-low	1	Touch screen with key guard and head pointer	20	20.55	19.65

cognition.

Levels of Cognition. Because we were exploring the range of cognitive ability needed to operate the robot arm, we separated "user selection" into a perceptual portion and a motor portion. The perceptual portion included recognizing the desired object on the flash card, finding the object on the shelf and finding the object on the display. The motor portion was how long it took for the participant to activate the desired target.

We hypothesized that the participants with lower cognitive ability would take longer for the perception time than the participants with higher cognitive ability. We computed unpaired, one-tailed t-tests on the perception times. We found that the participants with high cognition were able to perceive the desired object significantly faster than the other participants ($p<0.01$ for high cognition versus medium-high ($t(127)=4.64$), medium ($t(114)=3.42$), and medium-low ($t(85)=6.49$)). We also found that the participants with medium-high or medium cognition were able to perceive the desired object significantly faster than the participant with medium-low cognition ($p<0.01$ for medium-high versus medium-low ($t(80)=3.82$) and medium versus medium-low ($t(67)=3.38$)).

We found that 9 of our 12 participants selected the object before being told (i.e., during the perception portion) to do so at least once. The other three participants, P5, P7, and P12, did not because the display had to be moved into position before they could select.

We also computed unpaired, one-tailed t-tests on the motor time (i.e., the time of physical selection) for the participants who directly manipulated the touch screen with only their finger.[9] We found that the participants with high cognition were

[9]We excluded P5, P7, and P12 from the motor time (i.e., time to physically select the target) analysis. P5 used single switch scanning. P7 and P12 used a key guard over the touch screen; P7 selected with his

able to select the desired object significantly faster than the participants with medium-high cognition ($p<0.01$ with $t(93)=3.50$) and slightly faster than the participants with medium cognition ($p=0.07$ with $t(114)=1.80$).

For the perception time, there was no significant difference in performance between the participants with medium-high cognition and the participants with medium cognition. For the motor time, there also was no significant difference in performance between the participants with medium-high cognition who used their finger to point to the touch screen directly and the participants with medium cognitions. We believe that this is due to our broad definition of cognition used in this paper, which was comprised of several factors.

Incorrect User Selections. Overall, there were only 17 incorrect selections (91.4% correct). P5 made four incorrect selections, all in the first two sessions. In the first session, P5 made two incorrect selections because he was not wearing his glasses. With his glasses, P5 correctly selected the remaining two objects. In the second session, P5 incorrectly selected the first two objects. The experimenter reminded P5 that he should not rush and should take his time to select the objects correctly. P5 correctly selected the remaining two objects.

P7 made two incorrect selections due to his left side visual neglect. P7 had not yet learned to use head movements to extend his visual field, which was a skill that he was actively working on with his physical therapist. In both cases, the incorrect selection occurred at the end of a session as the difficulty increased. For the first incorrect selection, P7 knew what the object was and could see it on the shelf but not on the display. He selected incorrect objects twice and asked to select again. P7 selected the furthest left position on the key guard that he could see, which was actually only the middle of the key guard. For the second incorrect selection, P7 could not see the object on the left side of the shelf nor the display.

P12 made the most incorrect selections (ten out of twenty runs) due to his cognitive level, which was medium-low as described for this experiment. He did not always seem to understand that a specific target on the key guard was desired. Even with heavy prompting, P12 often merely chose the one that was closest and easiest for him to activate. We regard P12 as having the minimum level of cognition to use the robot arm.

Competence and Self-Esteem. We collected Likert scale ratings for a subset of the Psychological Impact of Assistive Devices Scale (PIADS) survey (37). We analyzed the participants' perceived competence (i.e., usefulness, independence, and confusion), adaptability (i.e., ability to participate, and eagerness to try new things), and self-esteem (i.e., happiness, sense or power, sense of control, and frustration). For each of these nine items, the participant rated from "strongly disagree" (-3) to "neither agree nor disagree" (0) to "strongly agree" (+3). Over all the sessions, participants rated their average competence as 1.74 (SD=1.00), adaptability 1.81 (SD=0.95), and self-esteem 1.36 (SD=1.00) which indicates an overall positive experience with using the robot. There was no statistical significance of the three ratings between the practice, first, and last sessions.

We looked at how the levels of cognition affects the participants' PIADS self-report. We computed one-tailed unpaired t-tests for competence, adaptability, and self-esteem. Due to the small number of participants, we do not wish to overstate the trends discovered in this fifteen week study. We found that participants with high cognition rated themselves to have higher competence than the participants with medium-high and medium cognition ($p<0.03$ with $t(19)=2.40$ and $t(17)=2.46$, respectively). Participants with medium-high cognition also rated themselves as

finger and P12 selected with his head pointer.

having higher self-esteem than the participants with medium cognition ($p<0.02$ with $t(16)=2.75$). We believe that one major factor in these results was the level of language ability. Participants with high and medium-high cognition were described as having good language ability and the students were fully or partially engaged in academic education.

6. Conclusions and Future Work

Our research has focused on replacing the default Manus ARM interface with a vision-based system while adding autonomy so that the robotic arm can execute a "pick-and-place" ADL. To this end, we have developed a simple image-based interface on which a person can touch or use a joystick to select an object. Further, we have developed a complete and autonomous object retrieval system.

We conducted a fifteen week study with twelve participants who had cognition ranging from high to medium-low. All participants were able to use the system and executed the 198 selections using their various access methods with only 17 incorrect selections (91.4% correct). We found that participants with higher levels of cognition were able to perceive the desired object faster than those with lower levels of cognition. Overall, the participants reported positive experiences using the system with respect to competence, adaptability, and self-esteem.

Our system was able to successfully retrieve an object from the bookshelf in 129 of the 198 trials (65% overall). Of the 69 unsuccessful retrievals, 56 (81%) were due to algorithmic failures, and the remaining 13 (19%) were due to the camera cables being disconnected as the robot arm moved during the trials. It should be noted that an unsuccessful retrieval does not mean the the robot system will never retrieve the object. User may repeat his/her selection.

We believe that the number of successful retrievals could be increased to 89% by integrating the cables into the robot arm, increasing the number of image views around each object, and using closed-loop feedback during gross motion of the cable-driven robot arm. To improve the gross motion (Algorithm 1), we believe that we can use color histogram analysis of the selected object for tracking as the end effector approaches the object which was used in a previous iteration of the robot system (1). We could also improve the alignment of the end effector on the object by continuously executing the visual alignment algorithms (Algorithms 2 and 3) while moving towards the object. To increase the success of object identification (Algorithm 4) and visual servoing (Algorithm 5), we could increase images views around each of the objects from three to five images like cardinal (north, east, south, and west) and ordinal directions (northeast, southeast, southwest, and northwest).

This system has not yet been optimized for speed or robustness. We are currently investigating online probabilistic estimation of depth information to limit the search region of local feature descriptors to make the fine motion visual servoing more robust. We will also design and analyze the stability of the region-based switched-control schemes (Algorithm 5) to reduce the time for the fine motion algorithm. Lastly, we need to investigate obstacle avoidance for when the desired object is occluded during gross motion.

We believe that this system can be used by people who use wheelchairs and have a limited area of reach and/or have limited upper limb strength. Our work has shown that the robot can be used by people have high to medium-high cognitive ability. The manufacturer's interface should only be used by people with typical cognition as per (8). Although our robot system is slow, we have shown that it can open up the range of people who can use the robot arm which will allow our target population to accomplish things they could not previously do independently.

We have shown that is it possible for a person with medium-low cognition to also use the system given the once per week interaction. With only once per week interaction, it may be difficult for people with lower cognition to remember how to use our system. We believe that as cognition decreases, the frequency of using the system must increase until the skill is mastered. We would like to conduct a long term case study in which the robot arm is used every day and perhaps multiple times per day by people with medium-low and low cognition.

7. Acknowledgments

This work was funded by NSF (IIS-0534364, IIS-0546309, IIS-0649736). The authors would like to acknowledge Dr. Hy Day and Dr. Jeffrey Jutai for the use of the Psychological Impact of Assistive Devices Scale (PIADS) (37). We would also like to acknowledge the use of the AT&T Labs Natural Voice®Text-to-Speech demo software (38).

The authors would like to thank the participants of Crotched Mountain Rehabilitation Center. We would like to thank Munjal Desai, William Harmon, and Adam Norton of UMass Lowell, and Kristen Stubbs of iRobot (formerly of UMass Lowell). We would like to thank Ryan Lovelett, Marcos Hernandez, Brandon Gilzean, Christopher Hamilton, and others at University of Central Florida. We would also like to thank the reviewers for their feedback.

References

[1] K. Tsui, H. Yanco, D. Kontak, and L. Beliveau, "Development and Evaluation of a Flexible Interface for a Wheelchair Mounted Robotic Arm," in *Human-Robot Interaction Conf.*, ACM, 2008.

[2] D.-J. Kim, R. Lovelett, and A. Behal, "Eye-in-Hand Stereo Visual Servoing of an Assistive Robot Arm in Unstructured Environments," in *IEEE Intl. Conf. on Robotics and Automation*, IEEE, 2009.

[3] International Center for Disability Information, "ICDI – The Total Number of Disabilities Reported by People Age 5 Years Old and Olders," 2003. `http://www.icdi.wvu.edu/disability/U.S\%20Tables/US7.htm`. Accessed Oct. 2009.

[4] H. Kayne, T. Kang, and M. LaPlante, "Disability Statistics Report: Mobility Device Use in the United States (Report 14)," 2000. `http://dsc.ucsf.edu/publication.php?pub_id=2`. Accessed Oct. 2009.

[5] Design in Motion, "The Problem with Manual Wheelchair Use," 2007. `http://www.dinm.net/freeflex.html`. Accessed Oct. 2009.

[6] Exact Dynamics, "iARM," 2010. `http://www.exactdynamics.nl/site/?page=iarm`. Accessed Jan. 2010.

[7] G. Römer, H. Stuyt, and A. Peters, "Cost-savings and Economic Benefits due to the Assistive Robotic Manipulator (ARM)," in *Proc. of the Intl. Conf. on Rehabilitation Robotics*, pp. 201–204, IEEE, 2005.

[8] G. Römer, H. Stuyt, and K. van Woerden, "Process for Obtaining a "Manus" (ARM) within the Netherlands," *Lecture Notes in Control and Information Sciences*, vol. 306, pp. 221–230, 2004. Part II: Rehabilitation Robots for Assistance of Human Movements, Chapter 14.

[9] R. Woodworth, "The Accuracy of Voluntary Movement," in *Psychology Review Monograph Supplement*, 1899.

[10] C. Stanger, C. Anglin, W. S. Harwin, and D. Romilly, "Devices for Assisting Manipulation: A Summary of User Task Priorities," *IEEE Trans. on Rehab. Engineering*, vol. 4, no. 2, pp. 256–265, 1994.

[11] B. Driessen, T. Kate, F. Liefhebber, A. Versluis, and J. Woerden, "Collaborative Control of the Manus Manipulator," *Universal Access in the Information Society*, vol. 4, no. 2, pp. 165–173, 2005.

[12] B. Driessen, F. Liefhebber, T. Kate, and K. Van Woerden, "Collaborative control of the manus manipulator," in *IEEE Intl. Conf. on Rehab. Robotics*, 2005.

[13] K. M. Tsui, K. Abu-Zahra, R. Casipe, J. M'Sadoques, and J. L. Drury, "Developing Heuristics for Assistive Robotics," 2010. Late breaking paper at ACM SIGCH/SIGART Human-Robot Interaction Conf.

[14] C. Dune, C. Leroux, and E. Marchand, "Intuitive Human Interactive with an Arm Robot for Severely Handicapped People - A One Click Approach," in *IEEE Intl. Conf. on Rehab. Robotics*, IEEE, 2007.

[15] C. Dune, E. Marchand, C. Collewet, and C. Leroux, "Active Rough Shape Estimation of Unknown Objects," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 3622–3627, IEEE, 2008.

[16] C. Dune, E. Marchand, C. Collewet, and C. Leroux, "Vision-based Grasping of Unknown Objects to Improve Disabled Persons Autonomy," in *Robotics: Science and Systems Manipulation Workshop: Intelligence in Human Environments*, 2008.

[17] Y. Choi, C. Anderson, J. Glass, and C. Kemp, "Laser Pointers and a Touch Screen: Intuitive Interfaces for Autonomous Mobile Manipulation for the Motor Impaired," in *Intl. ACM SIGACCESS Conf. on Computers and Accessibility*, pp. 225–232, ACM, ACM, 2008.

[18] H. Nguyen, C. Anderson, A. Trevor, A. Jain, Z. Xu, and C. Kemp, "El-E: An Assistive Robot That Fetches Objects From Flat Surfaces," in *Human-Robot Interaction Conf. Workshop on Robotic Helpers*, 2008.
[19] H. Nguyen and C. Kemp, "Bio-inspired Assistive Robotics: Service Dogs as a Model for Human-Robot Interaction and Mobile Manipulation," in *IEEE/RAS-EMBS Int. Conf. on Biomedical Robotics and Biomechatronics*, pp. 542–549, IEEE, 2008.
[20] J.-Y. Bouguet, "Camera Calibration Toolbox for Matlab," 2008. `http://www.vision.caltech.edu/bouguetj/calib_doc/`. Accessed Mar. 2009.
[21] K. Konolige, "SVS Development System," 2003. `http://www.ai.sri.com/~konolige/svs/svs.htm/`. Accessed Mar. 2009.
[22] Open Source Assistive Technology Software, "Dwell Click – oats," 2010. `http://www.oatsoft.org/Software/dwell-click`. Accessed Jul. 2010.
[23] Intel, "Welcome - OpenCV Wiki," 2009. `http://opencv.willowgarage.com`. Accessed Mar. 2009.
[24] S. Inspecta and B. Ricard, "cvBlobsLib - OpenCV Wiki," 2009. `http://opencv.willowgarage.com/wiki/cvBlobsLib`. Accessed Mar. 2009.
[25] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[26] D.-J. Kim, Z. Wang, and A. Behal, "Motion Segmentation and Control for an Intelligent Assistive Robotic Manipulator," *IEEE/ASME Transactions on Mechatronics*, 2011. In submission.
[27] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 2161–2168, 2006.
[28] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communication of the ACM*, vol. 24, pp. 381–395, 1981.
[29] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
[30] M. Ozuysal, P. Fua, and V. Lepetit, "Fast Keypoint Recognition in Ten Lines of Code," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
[31] J. Chen, D. M. Dawson, W. E. Dixon, and A. Behal, "Adaptive Homography-Based Visual Servo Tracking for a Fixed Camera Configuration with a Camera-in-Hand Extension," *IEEE Trans. on Control Systems Technology*, vol. 13, no. 5, pp. 814–825, 2005.
[32] O. Faugeras, *Three-dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA: MIT Press, 2001.
[33] Y. Fang, A. Behal, W. Dixon, and D. Dawson, "Adaptive 2.5 D Visual Servoing of Kinematically Redundant Robot Manipulators," in *IEEE Conf. on Decision and Control*, vol. 3, pp. 2860–2865, 2002.
[34] C. Hagen, D. Malkmus, and P. Durham, "Levels of Cognitive Functioning," *Rehabilitation of the Head Injured Adult: Comprehensive Physical Management.*, pp. 87–88, 1979. Downey, CA. Professional Staff Association of Rancho Los Amigos Hospital, Inc.
[35] R. J. Cooper, "Assistive Technology - CrossScanner Single Switch Method to Operate ALL Software," 2009. `http://www.rjcooper.com/cross-scanner/index.html`. Accessed Oct. 2009.
[36] D. M. McNair, M. Lorr, and L. F. Droppleman, "Profile of Mood States," in *Educational and Industrial Testing Service*, 1992.
[37] J. Jutai, "Psychosocial Impact of Assistive Devices Scale (PIADS)," 2009. `http://www.piads.ca/9/index1.2.html`. Accessed Oct. 2009.
[38] AT&T Labs, Inc. – Research, "AT&T Labs Natural Voice® Text-to-Speech Demo," 2010. `http://www2.research.att.com/~ttsweb/tts/demo.php`. Accessed Jan. 2010.