VIBI: Assistive Vision-Based Interface for Robot Manipulation

Camilo Perez Quintero, Oscar Ramirez, Martin Jägersand*

***Abstract*— Upper-body disabled people can benefit from the use of robot-arms to perform every day tasks. However, the adoption of this kind of technology has been limited by the complexity of robot manipulation tasks and the difficulty in controlling a multiple-DOF arm using a joystick or a similar device. Motivated by this need, we present an assistive vision-based interface for robot manipulation. Our proposal is to replace the direct joystick motor control interface present in a commercial wheelchair mounted assistive robotic manipulator with a human-robot interface based on visual selection. The scene in front of the robot is shown on a screen, and the user can then select an object with our novel grasping interface. We develop computer vision and motion control methods that drive the robot to that object. Our aim is not to replace user control, but instead augment user capabilities through our system with different levels of semi-autonomy, while leaving the user with a sense that he/she is in control of the task. Two disabled pilot users, were involved at different stages of our research. The first pilot user during the interface design along with rehab experts. The second performed user studies along with an 8 subject control group to evaluate our interface. Our system reduces robot instruction from a 6-DOF task in continuous space to either a 2-DOF pointing task or a discrete selection task among objects detected by computer vision.**

I. Introduction

Assistive robotics can help people with movement disabilities. In particular, robot manipulators can benefit people with limited upper body mobility. However, despite much research efforts robot manipulation in general human environments is still an open problem [1]. State of the art rehabilitation robot arms try to duplicate human arm functionality by providing 6 or 7-DOF motion. Typically manufacturers provide a joystick interface or a customized input device based on the end user's disabilities. User typically suffer from disabilities such as spasms or muscular dystrophy making control a bigger challenge. Mapping from the high DOF arm to a 2DOF joystick requires switching between modes which couple the joystick to different translation, rotation and grasp motions. This is time consuming, cumbersome and increases complexity and cognitive load for users. While tele-manipulation is the norm in rehab, much robotics research focuses on autonomy. User interaction can be minimized by ceding more autonomy to the robot. However, as pointed out by Kim *et al.* [2], disabled users preferred to keep as much control as possible with the aim of reasserting their domain of interaction with their environment as well as to engage and exercise their cognitive abilities to the fullest.

*This work is supported by iCORE, NSERC and the Canadian Space Agency (CSA). We also thank Kinova robotics and Kary Wright for all their collaboration during this work.

Authors are with the Department of Computing Science, University of Alberta, Edmonton, AB, T6G2E8, Canada. caperez@cs.ualberta.ca

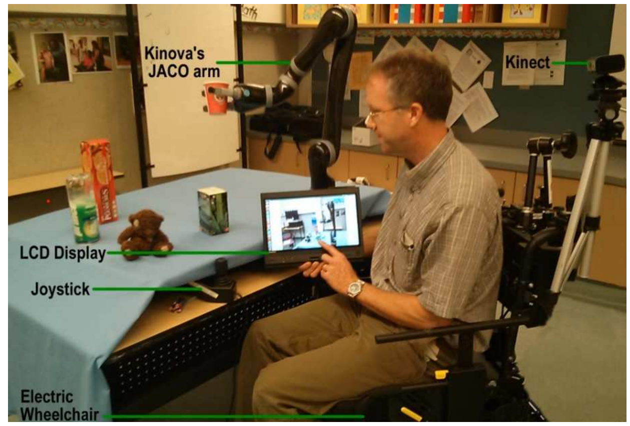

Fig. 1: Vision-based User Interface to a 6DOF robot arm and hand. By pointing and selecting in a 2D video image of the scene, the user can point to objects, select grasp types and execute robot actions. (Image edited from original to clarify that our system must be operated with both the arm and Kinect mounted on the wheelchair).

In the past decade several researchers have been working on simplifying the interaction between upper body disabled users and assistive robot arms, [3], [4], [5], [6], [7]. However, they focus more on how to achieve the manipulation task rather than how it is presented to the end-user. A more interesting interface implemented by [8] presents the operator with a 2D and 3D scene visualization and a wide range of interaction methods and tools to control a PR2. Numerous mobile manipulators have been used as assistive plataforms: PR2 [8], Care-O-bot 3 [9], EL-E [10]. A drawback with these systems, as pointed out by [8], is that despite previous advances in assistive mobile manipulators, none has been widely adopted by disabled end users to date. Part of the reason is their cost making them not suitable for widespread commercial adoption. In contrast, our system is evaluated on a Kinova Jaco, one of the most widely adopted robot arms by disabled users. Another limitation of mobile manipulators is the use of an allocentric system where the robot can move independently from the user. For instance Ciocarlie *et al.* [8] test their system with a pilot user completing a task that consists of retrieving a towel from the kitchen combining navigation and door/drawer opening and closing and object grasping. The complete task took 54 minutes. In our use-case, where the user is in constant use of a wheelchair and the manipulator, having the human-in-the-loop in a physical egocentric reference reduces operation time and solves many of the perception, navigation and manipulation problems encountered in other systems. The main strengths in our system compared to the others described above are adaptability and simplicity. Our interface turns the complexity of a multiple-DOF robot arm into a 2D object selection interface. Although our system is capable of doing autonomous grasping, our aim is to enhance user control rather than replace it by providing different levels of semi-autonomy our system can be easily adapt to different degrees of disability. We develop a computer vision-based system that focuses on user interaction. *Our main contributions are:*

- Intuitive 2D image-based interface (VIBI) with different levels of autonomy that hides the complexity of robot arms. Our HRI has evolved through several iterations of feedback from Kinova rehab engineers and from interactions with our first pilot user.
- Our system introduces a 2D grasping cursor, which allows the user to select a suitable grasp orientation.
- No need of a-priori 3D models of the objects, scene, and/or an image database.
- Easy portability from one robot-arm to another. Our system can be easily integrated with any robot arm that exposes basic functionality: angular joint control and current joint values. Our system was initially implemented on our lab's 7-DOF Barrett WAM arm, then ported to a 6-DOF Kinova Jaco.
- Quantitative and qualitative evaluation of our VIBI system through a user study with our second pilot user and eight control subjects.

II. System Description

Our proposed system uses a robot-arm, Kinect sensor, screen and a low power computer, see Figure 4. Typically, electric wheelchair users equipped with a robot-manipulator use the same wheelchair motion control device to interface with the robot-manipulator. This is adapted to ther capabilities by a physical therapist who selects an appropriate interface device for the handicap. Mouse emulation for PC operation is usually also designed and we take advantage of this for our interface. In our system, the scene in front of the user is shown on a screen, and the user can then select an object (see Figure 1). A demonstration of our system can be seen in the accompanying video.

We performed three iterations of revisions following feedback from a potential end user of our system (pilot user 1) and handicap robot engineers at Kinova. From this we learned: (1) Users prefer a system that works reliably with manual intervention to a system that is autonomous but not reliable. (2) Automating the 6DOF translation and rotation needed to reach and align for a grasp is a high value goal. (3) Users like to have direct control rather than autonomy when they perform a fine manipulation *e.g.,*grasp, poke object, drink.

A. Levels of Autonomy in the User Interface

As mentioned previously our aim is not to replace user control. This has led us to design our system with different levels of autonomy. We propose three types of object selection: **1D List Selection**, **2D Image Plane Selection**, **General Pick Selection** and one placing mode: **General Place** which are described below.

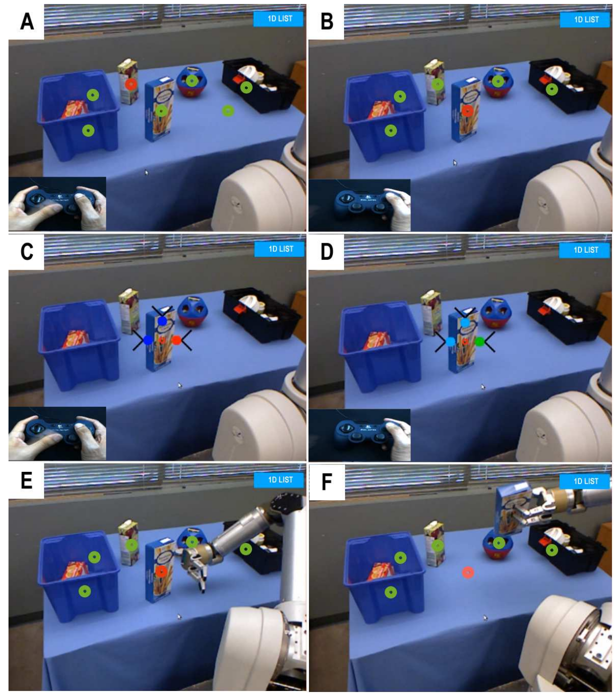

Fig. 2: 1D List Selection: Motion commands to the robot are given by discrete selection from displayed options (Gamepad is used to simulate user joystick input).
A. Computer vision detects objects on horizontal planes (tables, counters), generates a linear list of the objects and highlights the detected objects with green markers.
B. The user then iterates through the list to select an object. For users with spasms we threshold joystick vibration to trigger iteration.
C. A grasping cursor appears. The user can iterate again over the three possible gasping approaches: right, left and front.
D. Green or red indicates if the grasping is possible or not
E. The robot moves close to the selected object with the desired grasping orientation.
F. User can retake control of the robot and finish the grasping by joystick teleoperation or command the system to finish autonomously.

1) 1D List Selection: After interacting with our first pilot user who has spasms this control mode was designed for upper body disabled with high mobility restrictions. Discrete selection of the target object from a list requires users to generate noise with their input device in order to iterate over the available selections (see Figure 2).

2) 2D Image Plane Selection: This selection mode is intended for users with the ability to efficiently control the joystick coarsely in 2D. Instead of linearly iterating through the list of detected objects, the user selects from the 2D image. The system then highlights the detected object closest, in 3D, to the click and selects it (Figure 5A).

3) General Pick Selection: When the system is not able to detect an object in the scene correctly, or simply if the user wants a more flexible way to interact, **Pick Selection** allows an approximate positioning of the robot end-effector to any desired 3D location by having the user click in a 2D image and compute the corresponding 3D location. Refer to Figure 3A.

4) General Place Mode: After picking up an object, the user can select a scene location for placing it, *e.g.,* in Figure 3C the user wants to place a box of juice inside the blue container. In this case, the user clicks inside the blue container, and then selects a grasping approach by clicking in the grasp cursor. The robot-arm moves to an approximate location slightly above the desired location, with the selected grasping orientation. The user then finishes the task via teleoperation. In e.g., drinking and other proximal tasks it is common in handicap robotics to have pre-recorded poses. The user can also bring the object to any of these. Notice that the novelty of our interface relies on the simplicity of its visualization. Even though the point cloud output is available, what we present to the user is only a simple 2D RGB image, where we hide the complexity of the 3D grasp orientations by discretizing them.

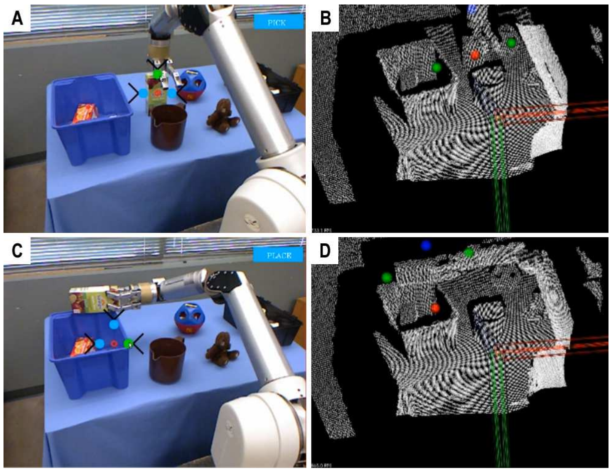

Fig. 3: Left column shows system user interface, right column a point cloud visualization.
A. User's view. The Pick mode is activated. The grasping cursor was used to position the robot from a top grasping approach.
B. Point Cloud view. The red sphere corresponds to the 3D location of the selected point in the RGB image. The blue sphere to the top grasping location, and the green spheres the right and left grasping locations.
C. The Place mode is activated. The grasping cursor is used this time to place an object. A right placing location is selected.
D. The red sphere corresponds to the 3D location of the selected point in the RGB image. The blue sphere to the top placing location and the green spheres to the right, left placing locations. Notice that the only difference between Pick and Place modes is the distance between the green, blue and red spheres.

B. System

Our system is composed of 4 modules as shown in Figure 4: **Interface**, **Calibration Vision** and **Robot-Arm** modules:

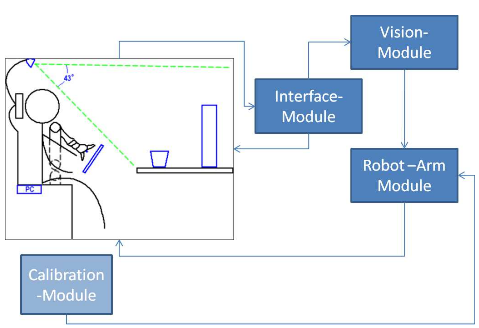

Fig. 4: System Diagram. Our proposed system uses a Kinect camera, screen and regular Linux PC. The Vision and Robot-Arm modules abstract the complexity of the arm which is presented to the end-user through the Interface module as a 2D image interface.

1) Interface: The person views the regular 2D RGB video from the Kinect scene camera on a screen. All user interaction is defined with reference to this visual interface. A blue label in the top right of the interface indicates the current mode (Figures 3 A and C) and is also used to change the current mode. In the case of the **1D List Selection**, the blue label (mode changing) can be included in the iterative selection to avoid clicking.

2) Calibration: Our system assumes that the RGB-D camera is fixed on the same reference frame as the robot-arm (*i.e.,*the arm and the Kinect are mounted on the wheelchair). We designed a visual 2D interface for calibration purposes. The arm moves the end-effector through a set of predefined points $R_i(x, y, z)$ corresponding to motions along the robot-arm frame of reference. The user then clicks on a known marked point on the robot at each of the i locations on the 2D image. Using the depth data from the Kinect the transformation matrix relating the Kinect's and the robot's frames of reference is acquired.

3) Vision: This module provides all the vision functionality using the Kinect camera which is composed of a depth and RGB camera.

Two approaches are used to locate the objects: (1) By using the depth and RGB camera correspondence, the user can directly select a 2D image region and the 3D coordinates associated with it is used as the estimate object location. This approach is used by **General Pick**, **General Place** modes, see Figure 3. When using these modes, the system provides a step trajectory tool (see Section II-B.4) where the user can stop the movement before any collision. (2) The system detects automatically the objects location in the horizontal plane closest to the Kinect. Both **1D List Selection** and **2D Image Plane Selection** use this approach. First the point cloud obtained from the Kinect is downsampled with a voxelized grid approach. Then using RANSAC [11] and a 2D convex hull, the table plane coefficients, inliers and points that belong to the table are found. Next, the inliers are clustered by distance to distinguish the objects. Finally, the mean vector for each cluster and the minimum bounding box are calculated.

After having the object's centroid and estimated bounding box, the grasp locations are calculated. The top grasp (blue sphere in Figure 5B) is calculated as a factor of the object bounding box height. To calculate the right and left grasp points, a grasping ring around the selected object (purple spheres in Figure 5B) parallel to the plane that holds it, is defined. Using the plane coefficients a normal vector $\mathbf{n} = \{n_x, n_y, n_z\}$ to the plane is calculated. Next the vector $\mathbf{a} = \{1, 1, \frac{-n_x - n_y}{n_z}\}$ parallel to the plane is found. Having $\mathbf{n}$ and $\hat{\mathbf{a}} = \{a_1, a_2, a_3\}$, where $\hat{\mathbf{a}}$ is the normalized vector of $\mathbf{a}$, we find $\mathbf{b} = \hat{\mathbf{a}} \times \mathbf{n} = \{b_1, b_2, b_3\}$. Based on these vectors, the location of the grasping spheres around the object centroid is defined by:

$$\mathbf{X}(\Theta) = \mathbf{c} + rcos(\Theta)\hat{\mathbf{a}} + rsin(\Theta)\mathbf{b}$$

The radius r of the ring is calculated based on the width value of the object's bounding box. The point cloud visualization of our system is shown in Figure 5B. The bear was selected, using the **2D Image Plane Selection** where the blue sphere and purple spheres around the bear were constructed as described above.

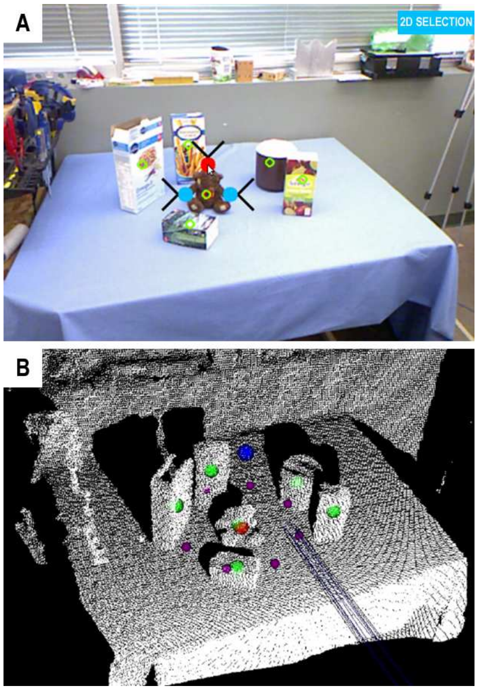

Fig. 5: **A.**The 2D Image Plane Selection is activated (blue label top right). Objects detected in the scene are marked by a green ring. The grasping cursor is being used to position the robot from a top grasping approach.
B.The green spheres are the 3D centroid locations of the objects detected in the scene. After selection is done, a red sphere appears representing the 3D location of the selected point in the RGB image. The system corrects the selection to the nearest centroid and constructs a ring of purple spheres used for the right and left graspings. A blue sphere is also added for the top grasping.

4) Robot Arm: Through this module, arm trajectories are generated given the grasp location and orientation calculated by the *vision* module. Arm trajectories can be executed completely autonomously, or iterated manually by the user. Additionally, it is possible to define constraints on regions within the arm's workspace. We use ROS and Moveit! [12], allowing for rapid deployment across different arms and enabling us to leverage work done by the robotics community. In order to add support for new arms within the system, we require a model of the arm and a simple interface to extract or set angular joint positions. In the case of the Jaco arm we later removed the ROS dependency and utilized the built in cartesian planer to reduce system requirements.

III. Experiments

Our system was tested by two disabled users and an 8 person control group. Our first pilot user was involved during our interface design along with rehab experts. He suffers from cerebral palsy, due to his condition, he has restricted upper limb movements and suffers from spasms. Feedback from Kinova and disabled users lead us to substitute the left grasp for a more common front grasp, and ROS MoveIt! for JACO's native motion planning. Although MoveIt! is more portable, it computes trajectories that users deemed unnatural and it is computationally expensive (Full day battery life is important). Our second pilot user has a form of quadriplegia. He has no hand or finger movements, just gross arm movements. He performed user studies along with an 8 subject control group to evaluate our interface. Our control group consists of 8 participants with normal hand-eye coordination aged from 18 to 34 with 6 males and 2 females. We performed two experiments for testing our system: an *orientation task* (Figure 6) and a *drinking task* (Figure 9). Both the pilot study user and our control group ran the two experiments. The average time per participant to complete the two experiments was 1 hour including break times and an initial 10 minute training period.

A. Tasks and data Analysis

The operator station consists of a mouse, screen and joystick. In front of the control station, a Kinect camera was located facing a 100x160cm tabletop with the JACO arm attached to the table. We adapt a joystick to the limitations of our disabled user and integrate a mouse pad that he uses on daily basis for internet navigation (see Figure 7).

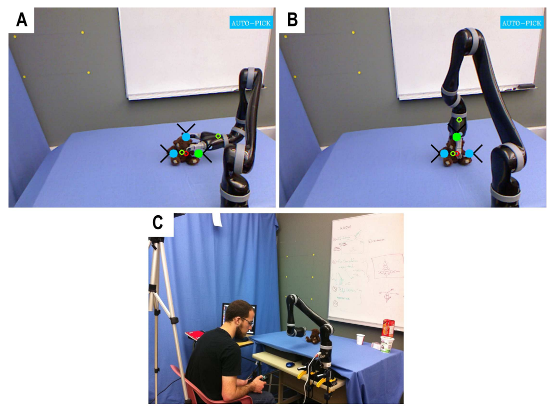

Fig. 6: Orientation Task. User was asked to locate the robot hand in a pre-grasp position for three specific orientations right (A), top (B) and front (C).

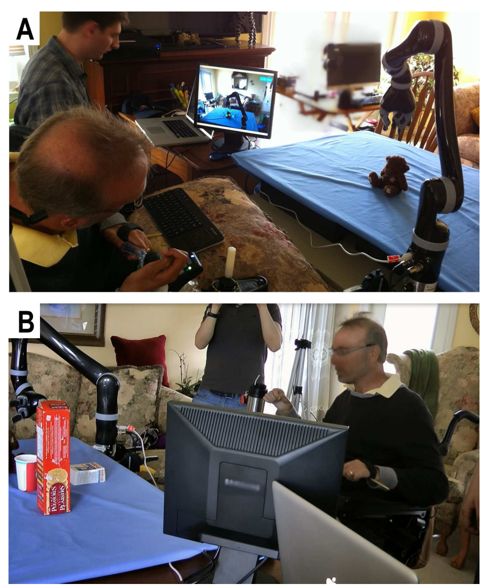

Fig. 7: Set-up for our pilot study. For direct teleoperation, we adapted a joystick. For interacting with the visual interface, we integrated end-user's regular PC mouse-pad. (A) Orientation task, (B) Drinking task.

1) Orientation task: The first experiment is a pre-grasp orientation task. An object was placed over tabletop and the

Orientation	t	P-value	Range
Right	2.73	0.018	(1.06, 19.37)
Top	4.14	0.004	(8.74, 34.05)
Front	4.85	0.004	(9.09, 27.66)

TABLE I: Control group orientation task: t, P-value and range of time improvement with 95% confidence. Range time in seconds.

user was asked to locate the robot hand in a right, top and front orientation with respect to the object (see Figure 6). Two control methods were used: (1) Direct teleoperation through JACO's built in Cartesian controller driven by a joystick and (2) Visual interface selection mode. To avoid bias, order of selection of the control method and orientation order was randomized for each user. During the task, the user was asked to complete 9 grasp orientations. During the experiment, we record the time it takes the user to complete each orientation and the number of times the user switches modes with the joystick to achieve translation, rotations and grasp motions of the arm. A comparison between the average time to complete the different orientations using the joystick and our proposed visual interface is shown in Figure 8. In the three orientation cases, the visual interface outperforms the joystick, for both the pilot study and the control group. The easiest orientation was right followed by front and top. This was expected because the JACO arm was mounted as a right handed one, where its starting position is a right orientation.

For the control group, a paired t test for comparing the two modes for each orientation was performed. We wanted to know if the interface decreases operation time as compared to the joystick mode, *i.e.,*$H_0 : \mu_d = 0$ versus $H_a : \mu_d > 0$ with a significant level $\alpha = 0.05$. Here $\mu_d = \mu_{Joystick} - \mu_{Interface}$ where $\mu_{Joystick}$ and $\mu_{Interface}$ are the mean time to complete the orientation using the joystick and using the interface respectively. The t and P_{values} for each orientation are shown in table I. Since $P_{value} \leq \alpha$ we reject H_0 : for all the orientations. This data analysis confirms that the interface mode decreases operation time in comparison with the joystick mode. A paired t confidence interval is also computed. Pairing the samples, the interval is given by $\bar{x}_d \pm (tCriticalValue) * \frac{S_d}{\sqrt{n}}$, where S_d is the standard deviation and n is the number of samples. Using a 95% confidence interval, the last column in table I shows the average range improvement in seconds between the joystick teleoperation and our interface (*i.e.,*we can be 95% confident that grasping from the top with our interface saves between 8.74 to 34.05 seconds in comparison with the joystick teleoperation). On average, our interface was faster for positioning and orienting the robot than the joystick interface. Disabled people in general have more difficulty providing input as seen in the joystick teleoperation performed by the disabled user, as shown figure 8. However since our visual user interface reduces the amount of input compared 6DOF direct robot control, the same task time can be reduced in some cases by a factor of 5 (compare in Figure 8, disabled user visual interface time with disabled user joystick time). Similarly the gap between the disabled user and the control group is also reduced.

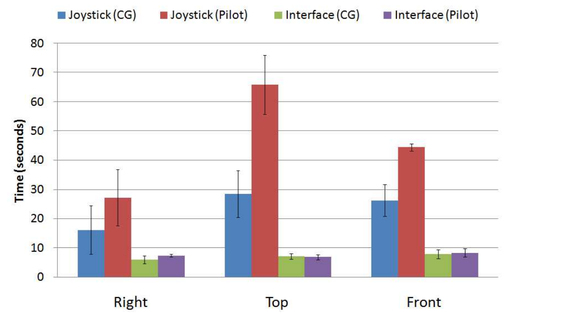

Fig. 8: Average time to complete orientation. When performing an orientation task our pilot study user was faster by 73%, 89% and 81% when using our interface to orient from the right, top and front respectively. When performing an orientation task the control group users were faster by 63%, 75% and 70% when using our interface to orient from the right, top and front respectively.

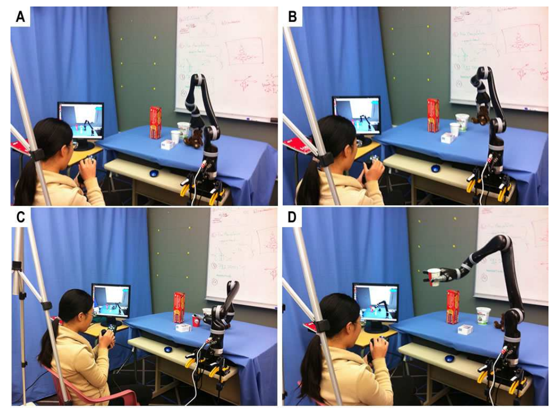

Fig. 9: Drinking Task. A cup is surrounded by obstructing objects, the user was asked to bring the cup to a position where he can drink from it. In this particular trial the user decides to moved first the bear to unblock the cup (A) and (B). Then, orient the hand in a right grasping position, pick up the cup and finally bring it close to her (C) and (D).

2) Drinking task: A cup was surrounded by different obstructing objects (see Figure 9). The user was asked to bring the cup to a position where she can drink from it. The user was also told that the cup was full of water and that her objective was to not spill the water during the process of bringing the cup close to her mouth. The user was free to decide what obstructions to remove and grasp to use. Two control methods were used. Each user performed the task three times per control method. To avoid bias, order of selection of the control method was randomized for each user. The two modes used were: (1) Control the arm with direct teleoperation through JACO's built in Cartesian controller driven by a joystick. (2) Alternate the arms control between direct teleoperation and an assisted control scheme where our visual interface was used to position the arm, and the joystick controller was used to finalize the grasping task. A comparison between the direct teleoperation using the joystick and the mixed joystick-interface approach is shown in Figure 10. The disabled user performed slightly better with joystick-interface than by only using the joystick. However, the control group performed better using only the joystick. A possible explanation of the results is that shifting between the interface and joystick may increase execution time. The better performance of the disabled user could be attributed to the higher demand that joystick control entails when faced with limited mobility.

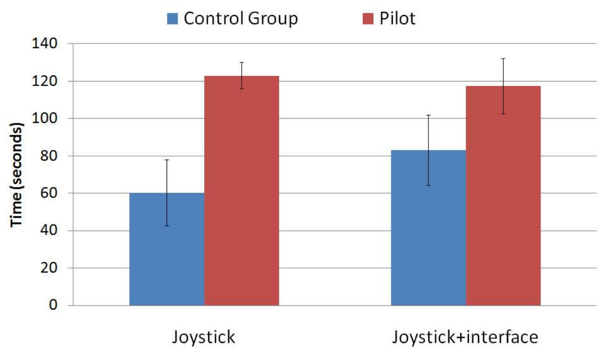

Fig. 10: Time to complete the drinking task for the joystick and a mixed mode using the joystick and our proposed vision interface.

3) Subjective analysis: At the end of the experiment, users were asked to fill a questionnaire. In the first section, the user rates in a Likert-type scale from 1 to 7 the difficulty of completing both tasks with the two different interfaces. The results are shown in Figure 11. Both tasks where completed by all users during our trials. During the *orientation task*, both disabled user and control group perceived the joystick at least 2 times more difficult to use than the interface. In the *drinking task*, the joystick-interface was perceived easier than the only joystick control mode. In general, users perceive the use of the interface easier than the direct joystick teleoperation. Something interesting to notice is that, although the time performance in the control group was better using only the joystick in the drinking task, the user perceived less difficulty when the interface was used. Thus indicating that faster performance does not necessarily reflect a better system.

The last question consists in rating how physically and mentally demanding the use of both interfaces was. The result is shown in Figure 11. It is clear that the use of our vision system is less demanding physically and mentally for both the disabled user and the control group. This is also expected because, as we mentioned, through our 2D interface, we hide the complexity of controlling a robot-arm.

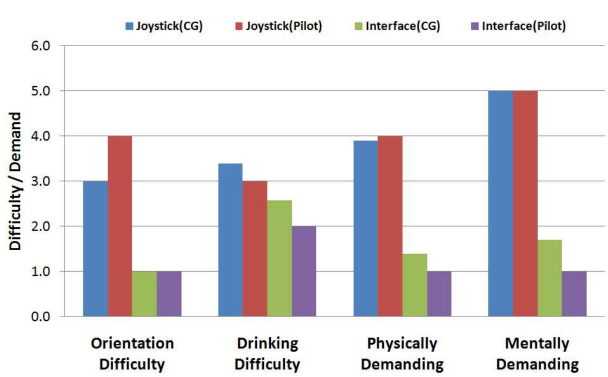

Fig. 11: Users rated on a scale from 1-7. Left: the difficulty of completing the orientation task and the drinking task using the joystick and our vision interface. Right: The physical and mental demand on completing both orientation and drinking task

IV. Conclusions

We designed and developed a computer vision system aimed to allow upper body disabled people to use a robot arm. Our system was implemented on two different robots: Barrett's WAM arm and Kinova's JACO arm. Our system was successively improved by having discussions with Kinova robotics and one disabled robot arm user who owns a Jaco and uses it daily. The user study was performed by a different disabled pilot user and 8 able participants in a control group. Our vision system on average was faster than the direct joystick control in achieving an orientation tasks. The vision system on average was 1.81 and 1.69 times faster than the direct joystick control in achieving orientation tasks for our pilot study and our control group respectively. While performing a drinking task, it was slightly faster in the pilot user study and slower in the control group. However, participants rated the vision system easier to use than direct joystick control of the arm for all cases. Our experimental results suggest that our system would be helpful to disabled users of wheelchair mounted robot arms such as Kinova's Jaco.

References

[1] C. C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 1, pp. 20–29, 2007.

[2] D.-J. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal, "How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot," *Systems, Man and Cybernetics, IEEE*, vol. 42, no. 1, pp. 2–14, 2012.

[3] B. Driessen, T. T. Kate, F. Liefhebber, A. Versluis, and J. van Woerden, "Collaborative control of the manus manipulator," *Universal Access in the Information Society*, vol. 4, no. 2, pp. 165–173, 2005.

[4] K. M. Tsui and H. A. Yanco, "Simplifying wheelchair mounted robotic arm control with a visual interface." in *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, 2007, pp. 97–102.

[5] K. M. Tsui, D.-J. Kim, A. Behal, D. Kontak, and H. A. Yanco, "i want that: Human-in-the-loop control of a wheelchair-mounted robotic arm," *Applied Bionics and Biomechanics*, vol. 8, no. 1, pp. 127–147, 2011.

[6] C. Dune, E. Marchand, C. Collowet, and C. Leroux, "Active rough shape estimation of unknown objects," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3622–3627.

[7] C. Dune, A. Remazeilles, E. Marchand, C. Leroux *et al.*, "Vision-based grasping of unknown objects to improve disabled people autonomy." in *Robotics: Science and Systems Manipulation Workshop: Intelligence in Human Environments.*, 2008.

[8] M. Ciocarlie, K. Hsiao, A. Leeper, and D. Gossow, "Mobile manipulation through an assistive home robot," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5313–5320.

[9] U. Reiser, T. Jacobs, G. Arbeiter, C. Parlitz, and K. Dautenhahn, "Care-o-bot® 3–vision of a robot butler," in *Your virtual butler*. Springer, 2013, pp. 97–116.

[10] A. Jain and C. C. Kemp, "El-e: an assistive mobile manipulator that autonomously fetches objects from flat surfaces," *Autonomous Robots*, vol. 28, no. 1, pp. 45–64, 2010.

[11] M. Fischler and R. Bolles, "Random sample consensus," *Communications of the ACM*, vol. 24, no. 6, 1981.

[12] S. Chitta, I. Sucan, and S. Cousins, "Moveit!" *IEEE Robotics Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.