Small Object Manipulation in 3D Perception Robotic Systems Using Visual Servoing

Camilo Perez Quintero, Oscar Ramirez, Mona Gridseth and Martin Jägersand*

***Abstract*— Robot manipulation has benefited from the introduction of low cost depth sensors. Researchers now have the possibility of easily integrating 3D perception into their robots. Several researchers have presented robotic systems that rely on low cost depth sensors, capable of doing manipulation tasks of the pick and place type. Furthermore Kinect-like sensors are design platforms for human-robot interaction that allow roboticists to develop better human-robot interactions. However, current low cost depth sensors are not optimized for robotics, and normally their resolution and operation range are not suitable for performing small object manipulation tasks. In this paper we give a brief overview of two human-robot manipulation interaction systems that we have developed based on a low cost depth sensor and our approach based on visual servoing to achieved higher precision and address small object manipulation tasks.**

I. Introduction

Artificial Intelligence technology has not reached the point where a general manipulation system behaves similar to a human. However, for a human-robot manipulation interaction, solutions that have the human-in-the-loop have demonstrated to be reliable [1], [2], [3], [4]. The two challenges here are: (1) Find natural communication mechanisms for human-robot interactions during a manipulation task and (2) the completion of the manipulation task. We have developed two interaction systems based on a low cost depth sensor. Our systems are capable of naturally interacting with a human to allow control during a complete pick and place task. However, while testing our systems we noticed the need for small object manipulation and the poor performance present in our system in these cases.

In this paper we give a brief overview of our natural manipulation interaction systems and our approach based on uncalibrated visual servoing to deal with small object manipulations. Section II presents an overview of two novel human-robot manipulation interfaces that are currently under development. Next, in section III we examine the main obstacles encountered while testing our different interaction systems followed by an overview of the UVS approach we follow to counteract these issues. Finally in section IV several tasks are explored in order to further illustrate our approach towards small object manipulation and the extensions currently being added to our systems.

*This work is supported by iCORE, NSERC and the Canadian Space Agency (CSA).

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

II. Human-robot manipulation interaction

Although household environment are typically classified as unstructured environments, objects are usually located in well-defined locations, e.g., tabletops, drawers, shelves, floor, etc. Researchers have taken advantage of this information and developed heuristic approaches for robot grasping [5], [1], [2], [6]. A common approach used by researchers using a Kinect-like sensor, consists of several steps: (1) Detect the closest or a selected horizontal plane. (2) Extract inliers by removing every point in the point cloud except for the points inside the volume defined by the plane and a maximum object height threshold. (3) Cluster inliers. (4) Process clusters for object grasping. (5) Based on heuristics, propose a set of grasps and implement a voting mechanism for selecting the best grasp.

Both of our interaction systems use a similar approach. Our two human-robot manipulation interactions are:(1) Selecting by pointing and (2) Visual based interface for people with an upper body disability.

A. SEPO: Selecting by pointing

For the first scenario, we are developing a vision-robotic system capable of understanding human pointing. Trough the pointing interface, our aim is to develop a better understanding of the role and form of spatial pointing and gestures in communicating manipulation tasks in a household scenario.

Our robot system uses a Kinect as the input device to interpret human gestures and the robot arm to render gestures to the human. We assume that the human and the objects are in the field of view of our depth camera and that the objects are located in a plane of interest that the robot can reach. Figure 1 shows a point cloud visualization of our system, not seen by the user, but shown here to illustrate the system implementation. A user points to a desired object, notice the virtual red line that is generated using the red sphere inside the users head and the teal sphere inside the users hand (for a detailed explanation in the pointing algorithm refer to [7]. After the hit location is found (red sphere at the end of the virtual ray) the system corrects the hit location to the nearest object and proposes a possible grasping location on top of the object (dark blue sphere). Figure 1 shows the identified objects in the system (green spheres) and the grasping location (dark blue sphere). A complete sequence of the interaction is shown in Figure 2.

B. VIBI: Visual base interface for upper body disabled.

For the second scenario, we are developing a vision based interface for people with upper body disabilities that reduces

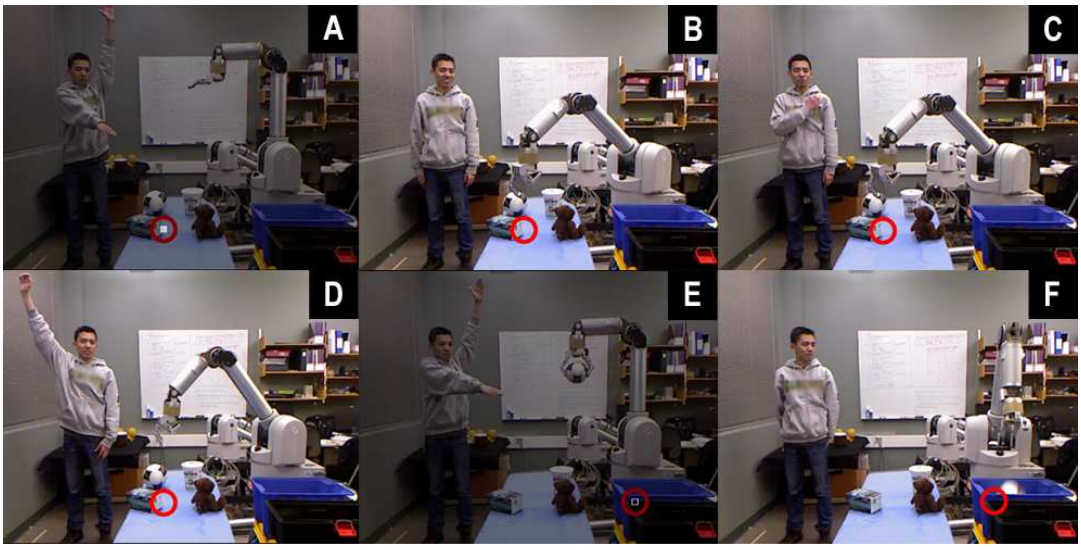

Fig. 2: A human-robot gesture sequence interaction with our system
A. Human instructor selects a desired object by pointing —- ”Pick the object that I’m pointing to”.
B. Robot assistant interprets the pointing gesture performed by the human and performs a confirmation gesture to the human instructor —- ”Is this the object that you want?”. Notice the pointing gesture performed by the robot arm.
C. Human instructor denies robot interpretation by crossing his dominant arm over his torso —- ”No”.
D. Robot assistant moves to the next possible selected object —- ”Is this the object that you point?”.
Human instructor confirms the robot interpretation by raising his dominant hand —- ”Yes”.
E. Robot assistant picks up the selected object. Human instructor selects a desired dropping location —- ”Drop the object in the blue container”.
F. Robot assistant places the object in the selected container.

robot operation complexity by providing different levels of autonomy to the end user (see Figure 4).

The user interface and point cloud visualization are shown in Figure 4. Our system is capable of detecting objects on top of a plane and by using a novel 2D image interface the user is able to select three grasping approaches (top, right and front) for autonomously picking and placing an object.

We have found experimentally that the SEPO system enables users to point to and select objects with an average position accuracy of 9.6 ± 1.6 cm in household situations [7] and in the VIBI system an average position accuracy of 3.0 ± 1.0 cm . Due to the sensor limitations we decided to explore a different approach by implementing a hybrid system using the Kinect sensor (3D perception) for large motions and two eye-in-hand cameras for achieving fine manipulations.

III. Hybrid approach for small object manipulation

During the evaluation of our systems, we have detected five factors that can affect grasping and/or interaction with small objects: (i) sensor range and resolution, (ii) end-effector, (iii) sensor localization, (iv) sensor-robot calibration and (v) control grasp approach.

(i) **Sensor range and resolution:** Current low cost depth sensors present an operation range between 0.8m and 5.0m which is not optimized for manipulation, where normally a range between 0 and 50cm is desirable. Sensor resolution restricts the size of possible objects to manipulate, for instance a straw, buttons, keys, or needles are hardly seen using the Kinect depth sensor.

(ii) **End-effector:** Similar to the vision sensor resolution, the end-effector size, shape and functionality limits the range of objects that can be handled and the robot manipulation capacity. Ideally a robot in human environments should have a similar grasping range to humans.

(iii) **Sensor localization:** Two sensor locations are commonly used in vision robot manipulation: (1) Eye-to-hand, where the robot end-effector is observed by a sensor (figure 5 left). Eye-in-hand, where the vision sensor is attached to the end-effector of the manipulator (figure 5 right). Both configurations have advantages and disadvantages. For instance the eye-to-hand configuration has a general overview of the complete scene, which is desirable for object detection and/or identification. However, for performing fine manipulations it is preferable to have a close look to the particular object from different points of view. The eye-in-hand configuration has the flexibility of positioning the vision sensor in a good viewing pose for manipulation, but after the object is grasped part of the sensor’s field of view can be obstructed. A good approach consists then in including both configurations on the robotic system. However, the eye-to-hand sensor will be normally fixed with respect to the eye-to-hand and if the eye-in-hand sensor is obstructed during the object manipulation the close view is lost.

(iv) **Sensor-robot calibration:** For small object manipulation calibration requirements are higher. A way to compensate error ranges is to use different sensors, with different error ranges. Although there are specific

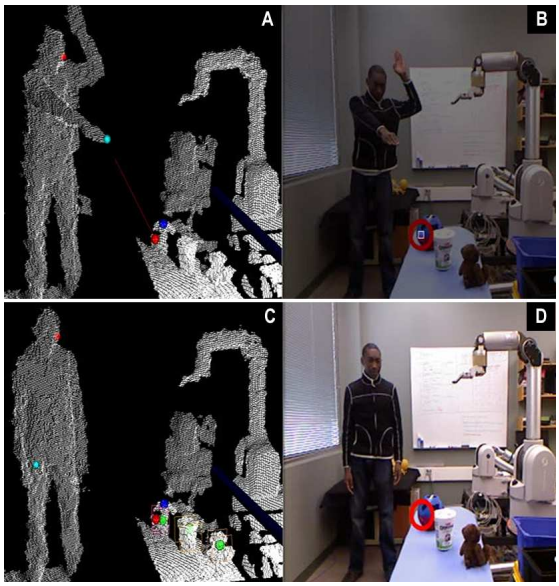

Fig. 1: System point cloud visualization (A,C) and RGB visualization (B,D): Human Selecting object by pointing (top). Red and light blue spheres attached to the human head and hand respectively (A). These two locations are used to find a virtual 3D ray (see 3D red ray between the hand and the hit point, A). Centroids (green spheres) and bounding boxes extraction from objects over the table plane (C). The system corrects the interpret hit point (red sphere) to the closest object (shape puzzle toy). The dark blue sphere above the selected object represents the correction done by the system and is used as the motion target for the robot finger when doing the confirmation gesture.

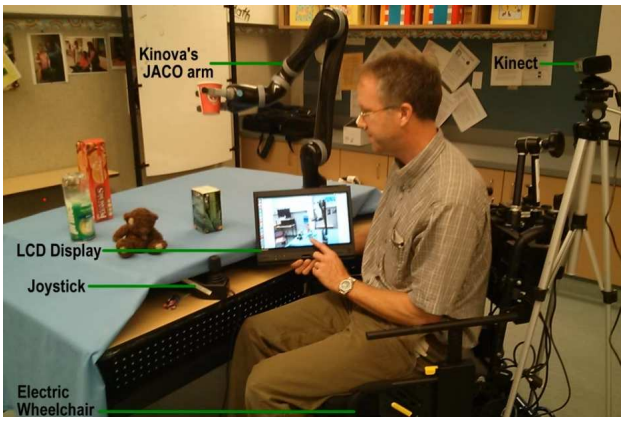

Fig. 3: Vision-based User Interface to a 6DOF robot arm and hand. By pointing and selecting in a 2D video image of the scene, the user can point to objects, select grasp types and execute robot actions. The system interprets the 2D pointing and selections with respect to the 3D scene geometry and objects, and plans the 6DOF motion trajectories. Thus the system relieves the user of the difficulty of the direct control of a high DOF robot arm and hand system.

calibration procedures (*e.g.*, camera to camera calibration, camera to arm registration), fusing different sensors in the same robotic system with different resolutions is a complex task, and normally will require periodic re-calibration procedures [8], [9].

(v) **Control grasp approach:** In the majority of systems based on a depth sensor, the grasping is performed in a open loop approach, 'looking' then 'moving'. The accuracy of the resulting operation depends directly on the accuracy of the visual sensor and the robot end-effector. This also assumes the scene will remain static further restricting the possible tasks to perform.

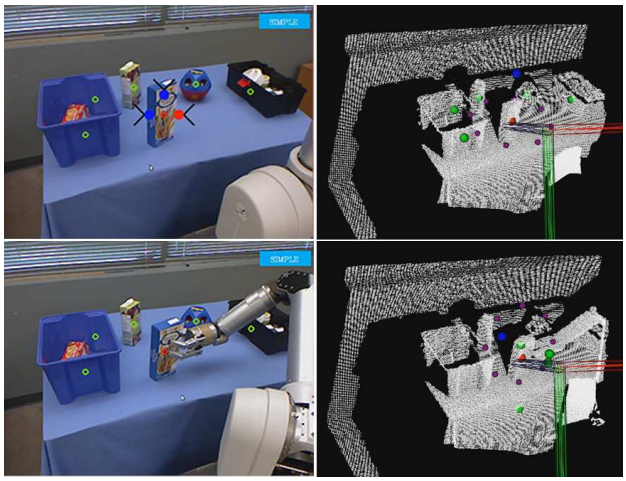

Fig. 4: Left column: User interface view. Right column: Point cloud visualization. The interface detects possible candidates for grasping represented as green rings in the user interface and green spheres in the point cloud visualization. The user selects a desired object and a grasping cursor pops-up (top-left). Through the grasping cursor the user selects a desired approach for grasping. In this example a right grasping approach was selected. Purple spheres in the point cloud visualization (top-right) show possible grasping candidates for the selected object. The user confirms the selection and the robot autonomously picks-up the desired object.

Fig. 5: Left: Eye-to-hand camera. Right: Eye-in-hand camera.

A. Approach

In this section we discuss and prototype some possibilities to address the above limitations of current 3D Kinect robot vision systems. A Kinect RGB and point cloud visualization are shown in figure 6. If our aim is to grasp the straw, some of the problems discussed in the above section are evident. The depth sensor resolution is not fine enough to retrieve the straw. The range of the sensor limits the sensor localization to a eye-to-hand approach, which for grasping the straw is not ideal. Also, if an open loop grasp approach is used it relies directly on the Kinect end-effector calibration.

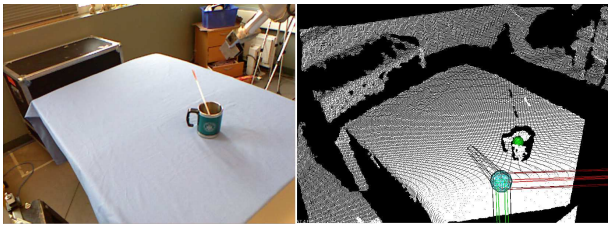

Fig. 6: Both RGB and point cloud visualization of a scene with a mug and a straw are shown. Notice that in the point cloud visualization the straw is barely observable.

Our improved approach consists of using the depth sensor

based system during general manipulations, where objects are in the resolution and range of the sensor. When objects are small and/or a more precise manipulation is required, we used the RGB-D system to approach and then a stereo eye-in-hand configuration is used to perform the fine alignment, see Figure 7. If the eye-in-hand cameras are obstructed after grasping the object, we used a similar approach to what Triggs *et al.* [10] proposed, where a second robot manipulator is used to obtain a good view for completing the manipulation task.

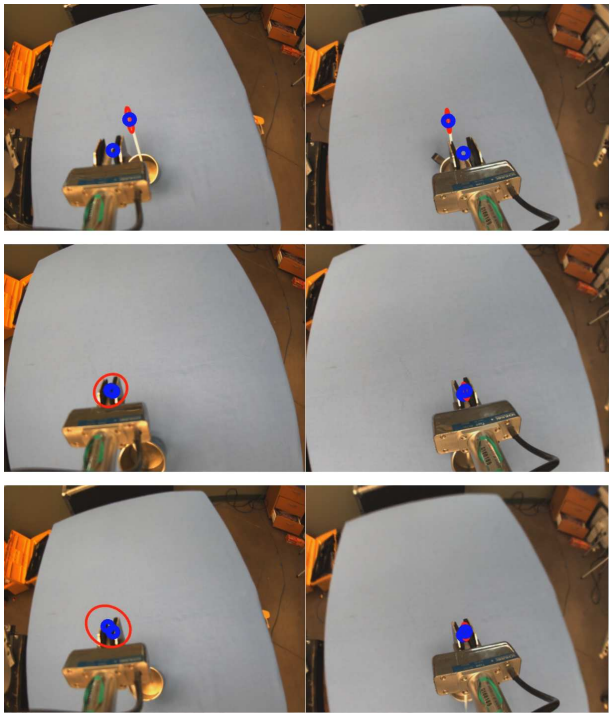

Fig. 7: Left column: left camera. Right column: right camera. The straw is tracked in both cameras, an error function between the two blue rings en each camera is formulated. When both errors are minimized the gripper is in position to grasp the straw.

Our approach for small object manipulation relies on uncalibrated visual servoing [11], which as opposed to our current system does not require calibration and the grasp is performed in a close loop approach. In the next two sections we review the principles of uncalibrated visual servoing and how a manipulation tasks can be specified in image space.

B. Uncalibrated Visual Servoing (UVS)

UVS should be defined without the need to reconstruct depth or other 3D parameters. See figure 8 for a UVS example. The relation between the image space s and robot joints q is commonly known as the visual-motor function:

$$\mathbf{s} = \mathbf{F}(\mathbf{q}). \tag{1}$$

A simple proportional control can be formulated as:

$$\dot{q} = -\lambda \widehat{\mathbf{J}}_{\mathbf{u}}^{+}(\mathbf{s} - \mathbf{s}^{*}), \tag{2}$$

where $\widehat{\mathbf{J}}_{\mathbf{u}}^{+}$ is the Moore-Penrose pseudoinverse of $\widehat{\mathbf{J}}_{\mathbf{u}}$ and s^{*} is the vector containing the desired values of the features. In the control law (2), the visual-motor Jacobian $\widehat{\mathbf{J}}_{\mathbf{u}}$ is estimated from data. In our approach we have chosen the Broyden method [11] to update the Jacobian due to its simplicity:

$$\widehat{\mathbf{J}}_{\mathbf{u}}^{(k+1)} = \widehat{\mathbf{J}}_{\mathbf{u}}^{(k)} + \alpha \frac{\left(\Delta \mathbf{s} - \widehat{\mathbf{J}}_{\mathbf{u}}^{(k)} \Delta q\right) \Delta q^{\top}}{\Delta q^{\top} \Delta q}, \tag{3}$$

where α is the learning rate applied to the rank one Broyden update. The initial estimate $\widehat{\mathbf{J}}_{\mathbf{u}}^{(0)}$ of the visual-motor Jacobian is obtained through small orthogonal exploratory motions.

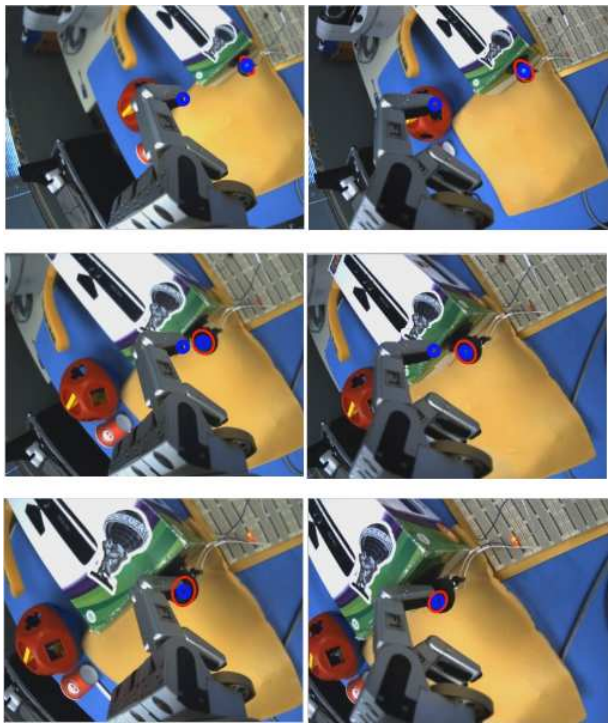

Fig. 8: The robot finger tip is marked in the image and the top of the button is tracked (top). With both pixel locations a point to point error is defined. Using the error in both images a visual servoing routine finds the correct movements to make the finger touch the button, a led is turned on when the finger presses the button (bottom).

C. Task Specification

For visual servoing to be useful we need to keep the 'human-in-the-loop' and let the user be able to specify tasks for the robot to complete. A visual task can be defined as the objective of bringing the end-effector of the robot arm to its target in the work space [12]. For example point-to-point and point-to-line tasks align a point on the robot end-effector with a point and line respectively in the workspace. These are simple tasks that can be specified in an interface by having the user click on end-effector and target points on the screen showing the robot camera view. Much of the work that a robot arm could do for the user requires more complex tasks. We can think of these complex tasks as being composed of several simple tasks. The interface should make it easy for the user to compose complex tasks out of simple ones. In the example in figure 9 we illustrate how one could

combine point-to-point and point-to-line tasks in two steps to align an end-effector with a hexagon and then close the gap.

The tasks that the user specifies in the image consist of 2D image information. Even though the end-effector converges to its goal in image space, that does not guarantee that the robot achieves the task in 3D world space. This means we need to be able to verify that the tasks specified by the user will in fact perform as expected. Depending on the level of calibration there exist tasks that can be unambiguously specified and verified using two 2D image views. They are said to be decidable tasks [12], [13]. For example point-to-point tasks are decidable for uncalibrated systems. It was shown by Dodds et al. [13] that there exists operators that can compose simple decidable tasks into more complex, but still decidable tasks. Therefore we can use simple decidable tasks to construct useful high-level tasks for the robot. We can create an interface that lets the user combine decidable tasks in an intuitive way to instruct the robot visual servoing.

Looking back at figure 9, we can get a set of equations that will allow us to use these tasks for visual servoing. The two point-to-line tasks between p_2 and L_1 and p_4 and L_2 will align the gripper with the hexagon and maintain the correct orientation throughout the process. The point-to-point tasks p_1 to p_5 and p_3 to p_6 will first bring the gripper close to the hexagon as seen in the developement from the left to the right image. Finally they will allow the gripper to close the gap to the hexagon so the grasp can be completed. We assume a stereo camera system, where L and R subscripts represent the left and right camera views respectively. Using image coordinates provided through the user interface the point-to-point and point-to-line tasks are represented as follows:

$$E_{pp}(p_1, p_2) = (p_{2L} - p_{1L}, p_{2R} - p_{1R})^T \tag{4}$$

$$E_{pl}(p, L) = (p_L \cdot L_L, p_R \cdot L_R)^T \tag{5}$$

where p_1, p_2, p and L represent the general point and line coordinates. We can combine the different tasks and preserve decidability by using the AND function defined in [13]. This allows us to stack the equations for several tasks that should hold simultaneously to give us the following error vector for the scenario in figure 9:

$$E = (E_{pp}(p_1, p_5), E_{pp}(p_3, p_6), E_{pl}(p_2, L_1), E_{pl}(p_4, L_2))^T \tag{6}$$

The vector E becomes the error function that we need to minimize in the visual servoing control law. Hence to complete the task we start with the situation in the left image and minimize E to servo the gripper to its location in the right image. Finally we update the locations of p_5 and p_6 and complete the task.

IV. Experiments

In experimental trials we have identified several scenarios where the use of UVS has allowed us to complete tasks that were otherwise unattainable through the use of the depth sensor.

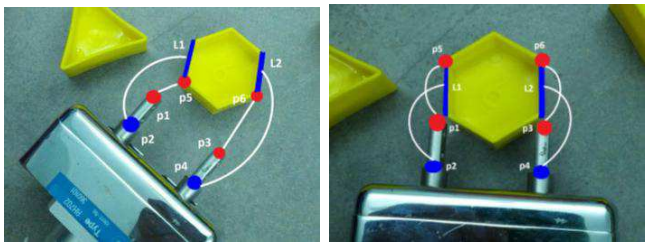

Fig. 9: Specification of point-to-point and point-to-line task to align a gripper with the hexagon and close the gap to grasp it.

As presented in Figures 6 and 7 the task of picking a straw from a cup can be challenging to a depth sensor as the size of the target is too small. Utilizing the eye-in-hand camera view point however the straw is big enough to facilitate the use of simple trackers. In our experiments we have found that a simple color tracker is sufficiently robust for completing the task.

A similar task involving a fishing lure where the goal is to thread a line through the loop of the lure is shown in 10.

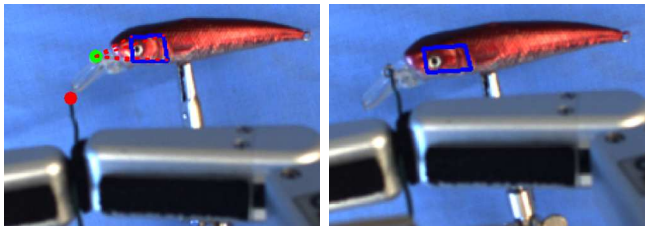

Fig. 10: Eye in hand view as the fishing lure task is completed. The end of the line is shown with a red circle, while the target is shown in green. The target loop location is tracked by suporting the point on the tracked patch of the head of the fish. Targets have been removed from the right image.

In this task the target point we wish to thread the line through is too small even for the eye in hand view. In order to complete the task we track the head of the lure and utilize the tracked region as a supporting structure for the desired target. This is shown in Figure 10. For the tracking we used the realtime registration-based tracking via approximate nearest neighbour search [14]. Given the corner points of the tracked patch and the target point on the lure, a homography is calculated and the target point is repositioned accordingly. The dimensions of the tracked patch on the head of the fishing lure are approximately $1.5\text{cm} \times 2\text{cm}$, the target whole is only 3mm in diameter. Given these dimensions it is not possible to approach the problem soley with a Kinect-like depth sensor.

Finally we explored the task of inserting a key in order to unlock a bike lock. The depth sensor failed to register most of the features of the lock given the resolution limitations of the sensor. The black matte finish of the lock also presented difficulties to the sensor. Utilizing a simple point to point constraint however we were able to place the key within the lock about half of the time. This task is facilitated in part due to the cone shape that is present in most keyholes. This mitigated the effects of the errors within our tracking. A view

of the point cloud for this lock is presented in Figure 11.

Fig. 11: Comparison of an eye in hand image of the bike lock and the corresponding point cloud.

Utilizing the aid of UVS we are able to successfully complete these tasks using simple trackers and point-to-point targets. By adding the task specification described in section III-C we hope to increase the robustness within our system for these tasks. For instance by performing a line to line alignment of the key to the key chamber before attempting to insert the key we could improve the robustness of this task.

Until now we have only explored the UVS approach while servoing the end effector in order to orient it accordingly for each task. The same mechanism can also be applied directly to the fingers of the robotic hand to further improve robustness of our system. This will be part of our future work.

V. Conclusions

We gave a brief overview of two human-robot manipulation interaction systems, that we are currently developing. By combining these perception systems with UVS we are able to perform grasps with a significant increase in precision on objects that were previously unmanageable. In future work we plan to further improve our system by allowing the definition of goals through the task specification framework proposed by Dodds *et al.* [13]. The challenge here is to find natural interaction mechanisms for the specification. With this augmentation users will gain the ability to define more precise goals through other error definitions like point to line or a combination of more user defined errors.

References

[1] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 1–8.

[2] 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.

[3] S. Muszynski, J. Stuckler, and S. Behnke, "Adjustable autonomy for mobile teleoperation of personal service robots," in *RO-MAN, 2012 IEEE*. IEEE, 2012, pp. 933–940.

[4] T. L. Chen, M. Ciocarlie, S. Cousins, P. Grice, K. Hawkins, K. Hsiao, C. C. Kemp, C.-H. King, D. A. Lazewatsky, A. E. Leeper *et al.*, "Robots for humanity: Using assistive robots to empower people with disabilities," 2013.

[5] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones, "Contact-reactive grasping of objects with partial shape information," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1228–1235.

[6] J. Stückler, R. Steffens, D. Holz, and S. Behnke, "Efficient 3d object perception and grasp planning for mobile manipulation in domestic environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1106–1115, 2013.

[7] C. P. Quintero, R. T. Fomena, A. Shademan, N. Wolleb, T. Dick, and M. Jagersand, "Sepo: Selecting by pointing as an intuitive human-robot command interface," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1166–1171.

[8] O. Birbach, B. Bauml, and U. Frese, "Automatic and self-contained calibration of a multi-sensorial humanoid's upper body," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3103–3108.

[9] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach," in *Experimental Robotics*. Springer, 2014, pp. 211–225.

[10] B. Triggs and C. Laugier, "Automatic camera placement for robot vision tasks," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 2. IEEE, 1995, pp. 1732–1737.

[11] M. Jagersand and R. C. Nelson, "Adaptive differential visual feedback for uncalibrated hand-eye coordination and motor control." DTIC Document, Tech. Rep., 1994.

[12] J. P. Hespanha, Z. Dodds, G. D. Hager, and A. S. Morse, "What tasks can be performed with an uncalibrated stereo vision system," *International Journal on Computer Vision*, vol. 35, pp. 65–85, 1999.

[13] Z. Dodds, G. D. Hager, A. S. Morse, and J. P. Hespanha, "Task specification and monitoring for uncalibrated hand/eye coordination," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1607–1613.

[14] T. Dick, C. P. Quintero, M. Jägersand, and A. Shademan, "Realtime registration-based tracking via approximate nearest neighbour search." in *Robotics: Science and Systems*. Citeseer, 2013.