2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Eye-in-Hand Stereo Visual Servoing of an Assistive Robot Arm in Unstructured Environments

Dae-Jin Kim, Ryan Lovelett, and Aman Behal

***Abstract*—We document the progress in the design and implementation of a motion control strategy that exploits visual feedback from a narrow baseline stereo head mounted in the hand of a wheelchair mounted robot arm (WMRA) to recognize and grasp textured ADL objects for which one or more templates exist in a large image database. The problem is made challenging by kinematic uncertainty in the robot, imperfect camera and stereo calibration, as well as the fact that we work in unstructured environments. The approach relies on separating the overall motion into gross and fine motion components. During the gross motion phase, local structure on an object around a user selected point of interest (POI) is extracted using sparse stereo information which is then utilized to converge on and roughly align the object with the image plane in order to be able to pursue object recognition and fine motion with strong likelihood of success. Fine motion is utilized to grasp the target object by relying on feature correspondences between the live object view and its template image. While features are detected using a robust real-time keypoint tracker, a hybrid visual servoing technique is exploited in which tracked pixel space features are utilized to generate translational motion commands while a Euclidean homography decomposition scheme is utilized for generation of orientation setpoints for the robot gripper. Experimental results are presented to demonstrate the efficacy of the proposed algorithm.**

I. Introduction

Over 6.8 million community-resident Americans use assistive devices as mobility aids. Two-thirds of mobility device users have limitations in one or more Instrumental Activities of Daily Living (IADLs) – this includes activities such as grocery shopping, using the telephone, meal preparation, light housework, etc. [1]. There are distinct groups that can be identified as demonstrating a requirement for assistance with mobility and manipulation. A very large group comprises individuals that suffer from neuromuscular diseases and injuries such as Spinal Cord Injury (SCI), Multiple Sclerosis (MS), Cerebral Palsy (CP), Stroke, Lou Gehrig's disease (ALS), etc. Many of these individuals are confined to wheelchairs, have moderate to minimal function in their upper extremities, and require some amount of attendant care [2]. Over the years, a variety of robotic assistive devices have been utilized to augment the functional capacity of the individual. WMRAs have been identified as a possible means to decrease reliance on the assistance of others in order to complete commonly performed ADLs [3]. Previous research has shown that a WMRA has the potential to increase activity level and social participation in individuals with Cerebral Palsy [4], Muscular Dystrophy [5], Spinal Cord Injury [6], and Multiple Sclerosis [7].

Of late, a lot of research has been focused on robust vision based controllers that are intended to remove the inherent choppiness and significant planning and cognitive load inherent with WMRAs that use either joint or Cartesian control mode. Even for able users, the cognitive effort in manipulating five or six directions (translation + rotation) is significant. Two vision based WMRA systems have recently been reported – the first work [8] exploits stereo vision from a fixed and an eye-in-hand camera and utilizes SIFT features combined with iterative Bayesian depth estimation to generate motion, however, it only performs gross motion (3D position) up to a textured object; the second work [9] requires the generation of 3D object models using SIFT [10] features which is combined with position-based servoing to generate the required robot motion to grab a textured object. Although not technically a WMRA, the companion robot El-E performs overhead object grasping off of "flat surfaces" using 3D laser scanning and vision [11]. Novel object grasping has been proposed in [12] using a supervised machine learning approach (trained on labeled synthetic data) that computes features related to edge, texture, color, and scale information – the 2D features obtained are then refined using features generated from 3D information obtained from a specialized range detector based on structured light.

In this paper, we present a motion control strategy that exploits feedback from a narrow baseline stereo camera system mounted in the hand of a WMRA to recognize and grasp textured ADL objects for which one or more templates exist in an image database. The problem is challenging because grasping for a desired object is required to be performed in unstructured environments (natural scenes) and in the presence of multiple nearby objects as well as occlusion and perspective distortion. Our approach relies on separating the overall motion into gross and fine motion components. During the gross motion phase, local structure on an object around a user selected point of interest (POI) is extracted using sparse stereo information (derived from matched SIFT [10] features) which is then utilized to converge on and roughly align the object with the image plane. The idea behind gross motion is to increase object resolution (through approach motion) and decrease perspective distortion (through alignment with a local

This study was funded by the National Science Foundation grant #IIS-0649736.

D.-J. Kim is with the NanoScience Technology Center, University of Central Florida, Orlando, FL 32826 dkim@mail.ucf.edu

R. Lovelett is with the School of EECS, University of Central Florida, Orlando, FL 32826 Ryan.Lovelett@gmail.com

A. Behal is with the School of EECS and the NanoScience Technology Center, University of Central Florida, Orlando, FL 32826 abehal@mail.ucf.edu

normal) which then synergistically combine to yield rich and reliable feature information for both object recognition and fine motion. For purposes of object recognition, stereo vision is utilized to statistically segment a pixel cloud in the live scene which is then run through a vocabulary tree [13] for fast matching with templates in a scalable database. False positives are rejected by way of utilization of a geometric constraint. To robustify against false negative matches, a PCA based analysis on a 3D point cloud roughly representing the object is developed to intelligently translate and orient the robot as needed. Finally, fine motion is utilized to grasp the target object by relying on feature correspondences between the live object view and its template image. While features are detected using a very robust real-time keypoint tracker based on ferns [14], we develop a variation of a hybrid visual servoing technique in which live pixel space features are utilized to generate translational motion commands while a Euclidean homography decomposition scheme is utilized for generation of orientation (yaw, pitch, and roll) setpoints for the robot gripper. An *ad hoc* scheme is utilized to ensure that the object does not exit the camera's field-of-view (FOV).

The paper is organized as follows. Section II introduces our research problem and conventions used in the paper. Section III describes our overall approach including gross motion, object recognition, and fine motion. We conclude in Section IV with some experimental results.

II. Problem Statement

The research objective is to design a motion control strategy for end-to-end automated object grasping while using a wheelchair mounted robotic arm (WMRA) in an unstructured environment. The measurements available for this purpose are joint angle feedback from the robot as well as live video streams from an end-effector mounted stereo head. The problem is complicated by kinematic uncertainties in the robot due to gearing and transmission. Furthermore, we deal with everyday (ADL) objects in natural environments (i.e., variable illumination, background, etc.) that may be occluded by other objects in the vicinity or by virtue of their pose with respect to the end-effector. Also, we work with natural features which may or may not be found/tracked in successive frames in the live view. Finally, limited image resolution, limited FOV, imperfect intrinsic and stereo calibration, as well as lens distortions are the other impediments that need to be dealt with.

We begin by explaining the nomenclature and conventions utilized in this paper. Here, $\mathbf{x}_z$ denotes a 3×1 vector $\mathbf{x}$ represented in a coordinate frame $\mathscr{F}_z$ while $\mathbf{x}_{z,y}$ denotes its y^{th} component. $\mathbf{R}_z^{xy}$ denotes an element located in the x^{th} row and the y^{th} column of the rotation transform matrix $\mathbf{R}_z$. $\mathbf{R}_{a2b}$ denotes a 3×3 matrix between coordinate frames a and b that can be applied to a vector expressed in coordinate frame a in order to obtain its representation in frame b. As shown in Fig. 1, three coordinate frames are used to describe the task space as follows: (a) the world coordinate frame $\mathscr{F}_w$, (b) the end-effector coordinate frame $\mathscr{F}_{ee}$, and (c) the camera coordinate frame $\mathscr{F}_c$. One can transform vectors expressed in one coordinate to another by using the appropriate transformation matrices. As an example, a rotation transform from camera coordinate frame $\mathscr{F}_c$ to world coordinate frame $\mathscr{F}_w$ can be described using the following equation.

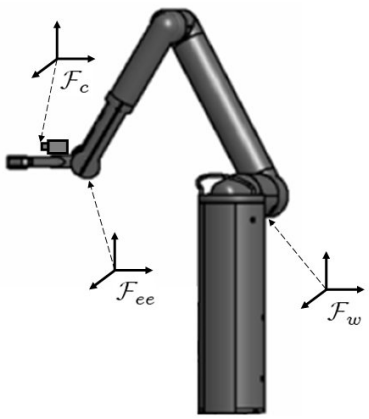


Fig. 1: Coordinate frames attached to robotic arm

$$\mathbf{R}_{c2w} = \mathbf{R}_{ee2w} \cdot \mathbf{R}_{c2ee} \tag{1}$$

Here, $\mathbf{R}_{c2ee}$ is a constant matrix determined by extrinsic camera-robot calibration process [15] while $\mathbf{R}_{ee2w} \triangleq \mathbf{R}_y \cdot \mathbf{R}_p \cdot \mathbf{R}_r$ can be computed using yaw, pitch, and roll angle feedback provided by the robot control box in Cartesian control mode – here $\mathbf{R}_y, \mathbf{R}_p$, and $\mathbf{R}_r$ denote standard yaw, pitch, and roll rotation matrices.

III. Approach

Fig. 2 provides a very succinct overview of the motion control strategy. A user can observe and determine the location of a target object through live video feedback provided inside of a GUI. By one of multiple user interface modalities (touchscreen, trackball, head tracker, speech, etc.), the user can indicate selection of a desired object. Stereo images of the scene are grabbed to determine average object distance from robot. If it is greater than a threshold, gross robot motion is initiated to zoom-in on and center the object as well as align the image plane with the object. Next, the object recognition module tries to find the best template image with the current grabbed image using a vocabulary tree; this is utilized in conjunction with RANSAC [16] and PCA based analysis in order to robustify against false positive and negative matches respectively. Finally, the ferns [14] based correct object model is loaded and a homography based hybrid visual servoing strategy is implemented to align the robotic hand accurately in front of the target object. Stereo is utilized to resolve the ambiguity resulting from the homography decomposition. When fine motion is finished, approach (motion along the axis of the gripper) and grab motions can be executed by the robot.

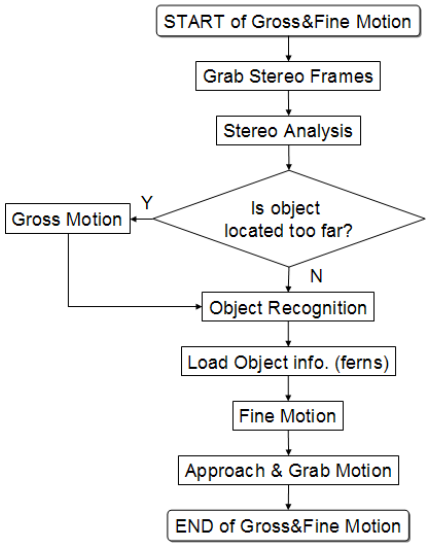


Fig. 2: Flowchart of the proposed Gross-to-Fine visual servoing strategy.

A. Gross Motion

1) Computing 3D information and normal vector: Gross motion visual processing is designed to allow the system to narrow attention within the vision system's wide FOV and then converge on the object to gather more resolution in the view and more disparity in the stereo. Preliminary testing in our laboratory showed that even segmentation of objects with known templates was unreliable without an initial gross motion – reasons were lack of resolution due to distance and perspective distortion due to steep viewing angles. Moreover, computation of a 3D point in front of the object and a normal to a locally approximated plane allows one to generate fast motion in the task space (no visual tracking is necessary which normally constrains speed). While such a scheme leads to errors in the final robot pose owing to kinematic uncertainty and errors in depth estimation, it is acceptable because these errors can be fixed during fine motion.

To get rough 3D position for the target object, we choose to use a blob-based local feature descriptor. Particularly, the SIFT descriptor [10] is chosen due to its invariance under scale, rotation, illumination, etc. An initial set of SIFT descriptors from left and right cameras (matched through epipolar constraints) is used to obtain a 3D point cloud around the user selected point on the GUI's live video feedback window. By using the extrinsic stereo calibration parameters $(\mathbf{R}_c^s, \mathbf{T}_c^s)$, we can setup the relationship between a 3D coordinate point on the left image $\bar{\mathbf{m}}_c^L$ and its estimated 3D coordinate point on the right image $\hat{\mathbf{m}}_c^R$ as follows

$$\hat{\mathbf{m}}_c^R = \mathbf{R}_c^s \cdot \bar{\mathbf{m}}_c^L + \mathbf{T}_c^s. \tag{2}$$

To minimize the estimation error $\mathbf{e} = \left\| \bar{\mathbf{m}}_c^R - \hat{\mathbf{m}}_c^R \right\|$ in a least-squared sense, the following equation is used to estimate depth

$$\mathbf{z}_c = \left(\mathbf{J}^T \cdot \mathbf{J}\right)^{-1} \left(\mathbf{J}^T \cdot \mathbf{R}_c^s \cdot \mathbf{m}_c^L\right) \tag{3}$$

where $\mathbf{z}_c = \begin{bmatrix} z_c^R/z_c^L & 1/z_c^L \end{bmatrix}$ with z_c^L and z_c^R denoting depth estimates in the left and right camera frames, respectively, $\mathbf{J} = \begin{bmatrix} \mathbf{m}_c^R & -\mathbf{T}_c^s \end{bmatrix}$, while $\mathbf{m}_c^L$ and $\mathbf{m}_c^R$ denote the measurable (given the intrinsic camera calibration matrices for the stereo head) homogeneous coordinates for the location of the keypoints in the left and right images, respectively. Thus, one can obtain a 3D point cloud from the stereo matching and depth estimation algorithms. Ideally, one expects the depth numbers from the left and right cameras to return identical values for meaningfully matched points; however, calibration errors cause the ratio to deviate from unity. Therefore, we remove incorrectly matched left and right feature pairs by computing the mean and variance of the depth ratios for all points and eliminating those with variance greater than a chosen threshold. Next, we remove 3D points that lie outside a physically meaningful region around the user's selection; in particular, considering the size of regular ADL objects, we choose a six cubic inch window. Finally, nearby objects in the background are removed by running statistics on the depths for the remainder of the points and removing those with high variance.

Given the 3D point cloud as obtained above, one expects to easily obtain a normal in the least square sense. However, depth estimation errors, our use of transparent and translucent objects as well as the sparseness of the data leads us to utilize a voting based algorithm to compute a robust normal vector. We take advantage of the fact that objects are either laid down or upright in indoor environments. Furthermore, the matched point cloud must originate from surfaces that are clearly visible from both cameras – thus, one can define an angular range about the optical axis of the camera. We discretize this feasible range and obtain N prototype normal vectors which we define in the world coordinate frame as $\mathbf{N}_w = \left\{ \mathbf{n}_w^1, \quad \ldots, \quad \mathbf{n}_w^N \right\}$. A prototype normal $\mathbf{n}_w^x$ and the 3D coordinates of one of the points $\bar{\mathbf{m}}_c^y$ are used to define a plane $\mathbf{P}_w^{x,y}$. Then, one can determine a set of inliers from the point cloud given the plane $\mathbf{P}_w^{x,y}$ and a Euclidean distance metric with a user defined threshold. One can then repeat to find inliers for each of the prototype normals. Based on the number of inliers, the normals are given a rank. This process of finding inliers and ranking the normals can then be applied for each point in the cloud. Finally, a histogram analysis is applied to obtain the normal with the most number of votes. In our system, we chose four normal vectors to represent three orientations of an upright object and one orientation for a laid down object. During the first matching process, geometry RANSAC is used to get rid of many outliers among all the possible matched pairs.

2) Motion Control: Given the estimated 3D location of the user's selection and the computed normal as shown above, one can compute the desired setpoints for position and orientation of the robot end-effector. While the normal allows for the computation of the end-effector yaw and pitch angles (normal is invariant to roll), the location of the end-effector is offset by a specifiable distance along the normal so that the end-effector comes to rest at a reasonable distance away from the object from which object identification and fine motion may

be eventually pursued. The setpoints for translation $\mathbf{p}_w^*$and rotation motions (yaw: $\theta_{w,y}^*$ and pitch: $\theta_{w,p}^*$) are computed by the following equations:

$$\begin{array}{rc} \mathbf{p}_w^* = & \mathbf{p}_w^t + \mathbf{p}_w^{ee} + \mathbf{R}_{ee2w} \cdot \mathbf{d}_{ee}^c \\ \theta_{w,y}^* = & \arctan\left(\mathbf{n}_w^{*,y}, \mathbf{n}_w^{*,x}\right) \\ \theta_{w,p}^* = & \arctan\left(\mathbf{n}_w^{*,z}, \sqrt{(\mathbf{n}_w^{*,x})^2 + (\mathbf{n}_w^{*,y})^2}\right) \end{array} \quad (4)$$

where $\mathbf{p}_w^t$ can be computed as follows

$$\mathbf{p}_w^t = \mathbf{R}_{c2w} \cdot \left(\mathbf{e}_c^t \cdot z_c^L - \mathbf{n}_c^* \cdot d_c^o\right). \quad (5)$$

Here, the vector $\mathbf{e}_c^t \triangleq \begin{bmatrix} (\mathbf{m}_c^{s,x} - \mathbf{m}_c^{o,x}) & (\mathbf{m}_c^{s,y} - \mathbf{m}_c^{o,y}) & 1 \end{bmatrix}^T$ denotes (in homogeneous coordinates) the error between the center point of image (constant) $\mathbf{m}_c^o$ and the user's selected point $\mathbf{m}_c^s$, d_c^o denotes the offset distance of the end-effector from the object, while $\mathbf{n}_c^*$ denotes the normal vector expressed in camera coordinates. Finally, translational/rotational velocity commands are generated based on a proportional controller.

B. Object Recognition and Fine Motion

While SIFT descriptors can be utilized to recognize objects, it becomes computationally intractable to extract the correct template through a brute force method especially when the database grows extremely large. To sidestep this issue, we utilize a vocabulary tree that provides for scalable recognition (SRVT) [13]. SRVT consists of a multi-level decision tree and visual keywords as leaf nodes. It is easily extendible and scalable to deal with lots of different natural scenes. We utilized more than 40,000 frames from action flicks to build our vocabulary tree. For purposes of our application, SRVT does not work very well with the raw scene obtained after gross motion. However, the use of stereo to further segment the scene and localize the object greatly enhances the discrimination capability of SRVT. RANSAC is utilized to eliminate false positive matches from the top five retrieved results while a PCA based analysis is run to reorient the end-effector and increase the likelihood of success if the initial SRVT and RANSAC processes fail to return a match.

Once the object is identified, fine motion can be executed. The goal for fine motion planning is to finely translate and align the gripper with the object in a grasping pose. During this phase, the kinematic uncertainty in the robot imposes a requirement for online feature detection and tracking. While SIFT is one of the most reliable features detectors, its disadvantage is the requirement for heavy computation time even with the most advanced multi-core processor. For this study, we adopt a fast and reliable feature descriptor based on ferns [14] for execution of fine motion.

1) Motion Control: A 2-1/2D (or hybrid) visual servoing scheme [17] is adopted to generate translation and rotation motion to align current eye-in-hand view with the pertinent template database image. After matching local descriptors of current image and loaded template image, the matched pairs are used to compute a homography between the feature locations. Next, the computed homography (Euclidean) is decomposed into two feasible solutions with appropriate rotation/translation motions. Here, one of the solutions is chosen by using the third view from our knowledge of an auxiliary stereo frame and the extrinsic calibration parameters of the stereo rig. Then, we can choose the best rotational transform matrix $\mathbf{R}$ to derive the axis of rotation $\mathbf{u}$ and rotation angle θ. For translation motion, one of the local descriptors is used as an anchor point $\mathbf{m}$ to track in $x-y$ plane of the camera coordinate frame. For approaching motion, the depth ratio (Z_1 and Z_1^* denote the depth information of current pose and final pose, respectively) is used to define an error signal as follows [18]

$$\log\left(\frac{Z_1}{Z_1^*}\right) = \log\left(\frac{(1 + \mathbf{n}^T\mathbf{x}_h)\mathbf{n}^{*T}\mathbf{m}^*}{\mathbf{n}^T\mathbf{m}}\right) \quad (6)$$

where $\mathbf{n}$ defines the computed normal on the object while $\mathbf{x}_h$ denotes a scaled distance between the current and final camera locations – we note here that both the normal and the distance vector are computed from the homography decomposition. For stable operation of fine motion, however, it is really important to keep all the features inside of FOV. Hence, during fine motion planning, a region-based switching scheme is applied to keep most of the features inside the FOV. A two stage approach is utilized. First, the target object is centered in the FOV to get higher resolution and large number of local descriptors which are extremely critical to get a higher confidence homography solution. Secondly, with centered object, real-time rotation/translation velocity commands are generated using homography analysis.

a) Phase-I – Centering of Target Object: This control law generates only translation velocity commands for $x-y$ plane. No approach/rotation motions occur at this step.

$$\mathbf{v}_w = \mathbf{K}_v \cdot \mathbf{R}_{c2w} \cdot (\mathbf{p}_c^e - \mathbf{p}_c^o) \quad (7)$$

where $\mathbf{p}_c^o$ and $\mathbf{p}_c^e$ stand for the center point of image (constant) and the centroid of found local descriptors on the current image after matching process with loaded template image, respectively.

b) Phase-II – Alignment with Target Object's Template: The control law for translation motion is almost similar with previous step except approach motion is added using the depth ratio computed by homography analysis. Instead of using the constant center point of image $\mathbf{p}_c^o$ and centroid of the remained local descriptors $\mathbf{p}_c^e$, the closest local descriptor among the remained points on the current image $\mathbf{p}_c^f$ and its corresponding point on the template image $\mathbf{p}_c^*$ are chosen to generate required velocity command as follows:

$$\mathbf{v}_w = \mathbf{K}_v \cdot \mathbf{R}_{c2w} \cdot \mathbf{e}_v \quad (8)$$

$$\mathbf{e}_v^{1,2} = \mathbf{p}_c^f - \mathbf{p}_c^*, \mathbf{e}_v^3 = \log\left(\frac{Z_1}{Z_1^*}\right) \quad (9)$$

Next, from the chosen rotation transform matrix $\mathbf{R}$ from homography decomposition, rotational velocity commands ω_{ypr}

are generated using setpoint information θ_w^* as follows

$$\theta_w^* = \begin{bmatrix} \arctan(\mathbf{R}^{21}, \mathbf{R}^{11}) \\ \arctan(\mathbf{R}^{31}, \sqrt{(\mathbf{R}^{32})^2 + (\mathbf{R}^{33})^2}) \\ -\arctan(\mathbf{R}^{32}, \mathbf{R}^{33}) \end{bmatrix} \quad (10)$$

Finally, an appropriate rotation transform matrix $\mathbf{R}_{ypr2w}$ is utilized to transform from the world coordinate frame $\mathscr{F}_w$ to the yaw-pitch-roll coordinate frame $\mathscr{F}_{ypr}$. Thus, ω_{ypr} can be computed as

$$\omega_{ypr} = \mathbf{K}_\omega \cdot \mathbf{R}_{c2ypr} \cdot \mathbf{R}_{ypr2w} \left(\theta_w^f - \theta_w^*\right) \quad (11)$$

where θ_w^f is updated in real-time from encoder feedback while $\mathbf{R}_{c2ypr} = \mathbf{R}_{w2ypr}\mathbf{R}_{c2w}$ where $\mathbf{R}_{w2ypr}$ can be computed as follows

$$\mathbf{R}_{w2ypr} = \begin{bmatrix} 0 & \mathbf{R}_{c2w}^{23} & \mathbf{R}_{c2w}^{13} \\ 0 & -\mathbf{R}_{c2w}^{13} & \mathbf{R}_{c2w}^{23} \\ 1 & 0 & \mathbf{R}_{c2w}^{33} \end{bmatrix} \quad (12)$$

In this step, to keep features within the current view, a simple switching scheme is used to limit the rotation errors. That is, when the x-y plane translation error is smaller than a user-defined error bound, the actual rotation error is used to create rotation motion of the robot. However, if the translation error is larger than a threshold, rotation error is set to zero to prevent any rotation motion of the robot. Hence, only translation motion is applied which allows the robot to maintain the feature set within the field of view. Because our fine motion relies on information from found local descriptors in each frame, this switching scheme is very important to make a robust tracking system.

IV. Experimental Results

In this work, the WMRA being utilized is the Manus ARM. The ARM has 6 + 1 (lift) + 1 (gripping) DOFs, a $80"(\simeq 1050mm)$ reach, maximum payload of $4.5lb(\simeq 2kg)$ and an exceptionally small footprint when folded in, thereby, allowing for an unobtrusive side or rear mount. For visual sensing of the environment, we utilize an end-effector mounted narrow baseline wide-angle stereo pair using standard surveillance video cameras PC223XP (dimension: $11mm \times 11mm$) housed in a mount that allows the generation of rectified stereo. Here, every single gray-level image is grabbed with PCI frame grabber with 320×240 pixel resolutions every $30ms$. Ordinary objects are used to test gross-to-fine visual servoing of MANUS arm. For rigorous testing, a variety of objects including upright and laid down objects as well as objects on a high table, a low table, and the floor are utilized.

As shown in Fig. 3, our gross-to-fine motion can successfully guide the robotic hand exactly in front of the water bottle on the high table. SRVT-based object recognition correctly retrieves a template image from the database. Fine motion (Phase-I(P-I) and Phase-II(P-II)) is performed via switching mechanism between image and hybrid visual servoing as seen in Fig. 3(e) and (f). As can be seen in Fig. 4, the robot can successfully grab a remote control from a low table (see (a)-(c)), a small and thin object like marker pen (see (d)-(f)) or a water bottle fallen on the floor (see (g)-(i)). See attached video clip of grabbing the remote control. Full version of video clip is also available at *http://www.eecs.ucf.edu/ abehal/AssistiveRobotics/*.

V. Concluding Remarks

This paper has introduced an eye-in-hand stereo visual servoing of a robotic arm in unstructured environments. Future plans are to test the developed robotic system first with healthy subjects at UCF and then with SCI patients at Orlando Health.

References

[1] A. Bookman, M. Harrington, L. Pass, and E. Reisner, "Family caregiver handbook," Massachusetts Institute of Technology, Tech. Rep., 2007.

[2] M. D. Association, "Neuromuscular diseases in the mda program," 1999, http://www.mdausa.org/disease/40list.html.

[3] S. D. Prior, "An electric wheelchair mounted robotic arm–a survey of potential users," *J. Med. Eng. Technology*, vol. 14, pp. 143–154, 1990.

[4] H. Kwee, J. Quaedackers, B. E. van de, L. Theeuwen, and L. Speth, "Adapting the control of the manus manipulator for persons with cerebral palsy: an exploratory study," *Technology and Disability*, vol. 14, pp. 31–42, 2002.

[5] J. R. Bach, A. P. Zeelenberg, and C. Winter, "Wheelchair-mounted robot manipulators: Long term use by patients with duchenne muscular dystrophy," *Am J Phys Med Rehabil*, vol. 69, pp. 55–59, 1990.

[6] S. L. Garber, A. L. Williams, K. Cook, and A. M. Koontz, "Effect of a wheelchair-mounted robotic arm on functional outcomes in persons with spinal cord injury," *Neurorehabil Neural Repair*, vol. 17, p. 244, 2003.

[7] G. Fulk, M. Frick, A. Behal, and M. Ludwig, "A wheelchair mounted robotic arm for individuals with multiple sclerosis: A pilot study," February 2009, submitted to Combined Sections Meeting of the American Physical Therapy Association.

[8] C. Dune, C. Leroux, and E. Marchand, "Intuitive human interaction with an arm robot for severely handicapped people - a one click approach," in *Proceedings of IEEE Int'l Conf. on Rehabilitation Robotics*, The Netherlands, 2007, pp. 582–589.

[9] F. Liefhebber and J. Sijs, "Vision-based control of the manus using sift," in *Proceedings of IEEE Int'l Conf. on Rehabilitation Robotics*, The Netherlands, 2007, pp. 854–861.

[10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] 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 *Proceedings of Human-Robot Interaction 2008 Workshop on Robotic Helpers*, Amsterdam, Netherlands, March 2008.

[12] A. Saxena, L. Wong, and A. Ng, "Learning grasp strategies with partial shape information," in *Proceedings of AAAI Confernece*, 2008.

[13] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2161–2168.

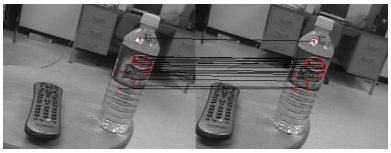
[14] M. Ozuysal, P. Fua, and V. Lepetit, "Fast keypoint recognition in ten lines of code," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Minneapolis, 2007.

[15] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3d roboticshand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, January 1989.

[16] 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.

[17] 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 Transactions on Control Systems Technology*, vol. 13, no. 5, pp. 814–825, September 2005.

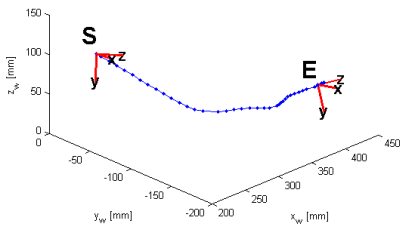
[18] Y. Fang, A. Behal, W. Dixon, and D. Dawson, "Adaptive 2.5 d visual servoing of kinematically redundant robot manipulators," in *Proceedings of IEEE Conf. on Decision and Control*, vol. 3, 2002, pp. 2860–2865.



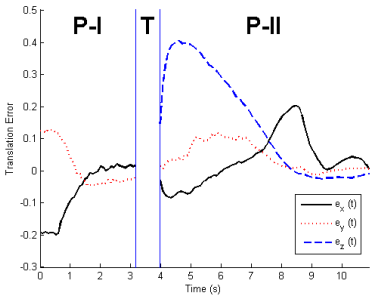
(a) Initial pose

(b) End of gross motion

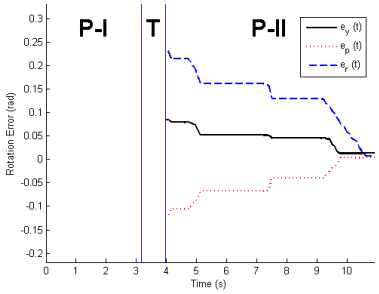
(c) End of fine motion



(d) 3D trajectory of gross motion



(e) Translation error in fine motion

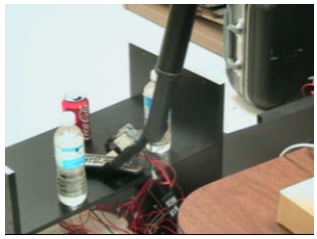


(f) Rotation error in fine motion

Fig. 3: Gross-to-Fine motion to grab a 'Nestle' water bottle on the high table

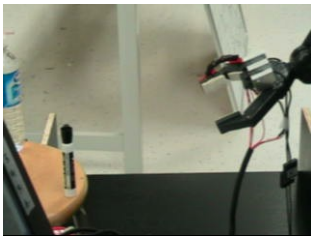
(a) Initial pose; remote control on the low table

(b) End of gross motion; remote control on the low table

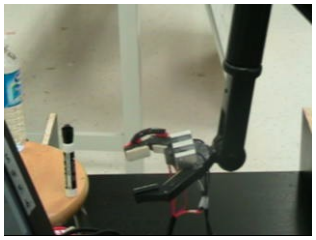


(c) End of fine & grab motion; remote control on the low table

(d) Fetching the object; remote control on the low table

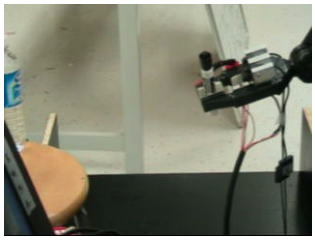


(e) Initial pose; marker pen on the high table

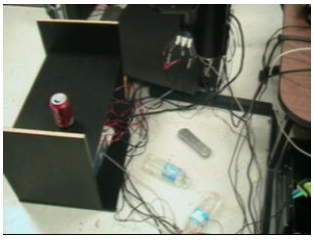


(f) End of gross motion; marker pen on the high table

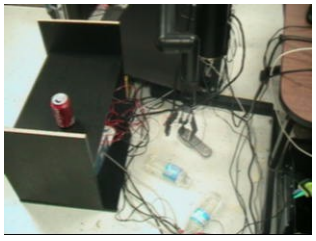
(g) End of fine motion; marker pen on the high table



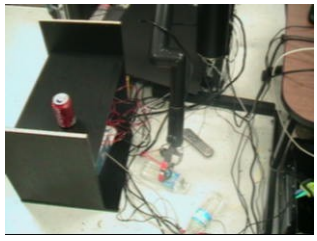
(h) Fetching the object; marker pen on the high table



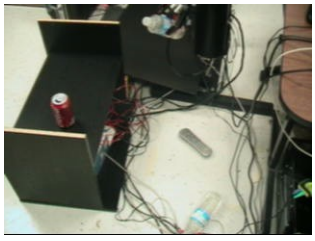
(i) Initial pose; water bottle on the floor



(j) End of gross motion; water bottle on the floor



(k) End of fine motion; water bottle on the floor



(l) Fetching the object; water bottle on the floor

Fig. 4: Gross-to-Fine motion to grab various objects