Stereo Image-based Visual Servoing Towards Feature-based Grasping

Albert Enyedy[1], Ashay Aswale[1], Berk Calli[1], Michael Gennert[1]

***Abstract*— This paper presents an image-based visual servoing scheme that can control robotic manipulators in 3D space using 2D stereo images without needing to perform stereo reconstruction. We use a stereo camera in an eye-to-hand configuration for controlling the robot to reach target positions by directly mapping image space errors to joint space actuation. We achieve convergence without a-priori knowledge of the target object, a reference 2D image, or 3D data. By doing so, we can reach targets in unstructured environments using high-resolution RGB images instead of utilizing relatively noisy depth data. We conduct several experiments on two different physical robots. The Panda 7DOF arm grasps a static target in 3D space, grasps a pitcher handle, and picks and places a box by determining the approach angle using 2D image features, demonstrating that this algorithm can be used for grasping practical objects in 3D space using only 2D image features for feedback. Our second platform, the Atlas humanoid robot, reaches a target from an unknown starting configuration, demonstrating that this controller achieves convergence to a target, even with the uncertainties introduced by walking to a new location. We believe that this algorithm is a step towards enabling intuitive interfaces that allow a user to initiate a grasp on an object by specifying a grasping point in a 2D image.**

I. Introduction

To reliably complete an everyday task such as picking up a book in a room to place on a shelf, a robot must operate often without knowing prior information about its environment. For example, today the book may be on a different table than what the robot has been calibrated to pick up from, or a differently-sized book may be used. Using cameras, the robot can see these differences in its workspace and use relevant image features to position its manipulator in the desired location relative to the book. Vision-based control can be used to complete such tasks, thus enabling a robot to grasp objects in unstructured environments using image features for feedback.

Robots can reach targets by using these image features in feedback control schemes such as visual servoing, in which motion in the image space is mapped to motion in the task space [1]. Visual servoing has two main categories: position-based visual servoing (PBVS) [2], [3], [4] and image-based visual servoing (IBVS) [5], [6], [7]; PBVS minimizes a 3D error vector between the task space location of the target and the end effector, which requires object models and 3D reconstruction. In this work, we focus on IBVS, which utilizes image features and does not require 3D reconstruction.

*This paper was supported in part by the National Science Foundation under grant IIS-1900953 and CMMI-1928506.

[1]A. Enyedy, A. Aswale, B. Calli, and M. Gennert are with the Department of Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA ajenyedy@wpi.edu

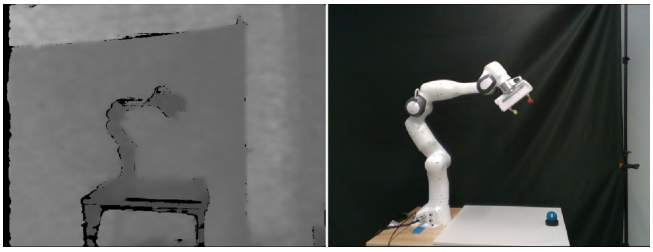


Fig. 1: Our visual servoing algorithm relies on RGB stereo imaging, and does not use depth data. In this way, we can achieve accurate control using high resolution feedback images, while avoiding noisy depth measurements. In this image, the target object (blue ball) is not even differentiable from the table in depth images due to noise, but can clearly be detected in RGB data.

The majority of existing IBVS approaches for grasping objects assume a given goal image (i.e. known locations for target features) and match the image features of the currently observed target object with the corresponding image features of its target position [8], [9]. However, in an unknown or unstructured environment, the robot may not have access to prior knowledge of the goal scene. These conditions would greatly increase the difficulty a robot would have with grasping an object, especially if the model of the target object is not known beforehand [10].

In this work, we utilize 2D RGB images for visual servoing for grasping applications; we do not require object models or a goal image (the target point is manually-selected) and do not rely on depth data or 3D stereo reconstruction. Our motivation stems from 2D images being much higher resolution, more reliable, and less noisy compared to depth images, as shown in Fig. 1. 3D stereo reconstruction is also prone to errors and inaccuracies, especially in low-texture environments [11].

Such purely 2D stereo visual servoing approach is proposed in the work of Hager *et. al.* [12] on robot hand-eye coordination. Our visual servoing approach implementation takes inspiration from Hager's work to grasp practical objects in 3D space. We set up two cameras side-by-side about 0.3 m apart looking at the robot arm, then we select a single point in each of the left and right camera images, and drive the robot manipulator to reach that point in 3D space to grasp an object. Furthermore, the pitch approach angle of the manipulator can be specified (but not the yaw or roll angles due to camera placement) and using only visual feedback we can grasp objects at desired gripper pitch approach angles. We believe that this approach is suitable for simple point-

and-click interfaces for robotics grasping applications. As such, it provides opportunities for streamlining control of robot manipulators in industrial robot arm cells by replacing complex teach pendants with a few clicks on a touchscreen. Regarding autonomous grasping, this approach can be coupled with 2D grasp synthesis algorithms to provide grasping points [10], and our system would require no additional sensors or data to reach the produced grasping target.

Summarizing our contributions and assumptions, we propose a stereo IBVS scheme that reaches and grasps objects without depth sensing, explicit 3D stereo reconstruction, or prior knowledge of the object model. We demonstrate its utility with simple point-and-click interfaces to grasp objects. We assume a rigid robot system with a known kinematic model. We also assume that both cameras in the stereo system are calibrated, and image features do not leave the cameras' field of view. When reaching target gripper orientations, we assume the gripper's fingers remain close to parallel to the camera's image plane, for the most accurate calculation of gripper approach angle in the 2D image. Our results show that the system can achieve convergence to grasp a static target at various depths from the camera, grasp the handle of a water pitcher, pick and place a practical object such as a keyboard box by specifying approach angles, and can reach a target from an uncertain starting position.

II. Related Works

The work of Maru *et. al.* [13] presents stereo IBVS control of a manipulator arm by positioning cameras on the end effector of the robot for a first-person view of the workspace. In this system, the pose of the stereo cameras can be used to calculate the image Jacobian during motion to map velocities from the image frame to the world frame. This camera setup limits the view of the workspace and arm itself as opposed to an eye-to-hand setup. The work of Hager *et. al.* [12] presents the concept of using the disparity between stereo cameras viewing the end effector and target object simultaneously in order to minimize the error, with robustness to calibration errors. Once the visual disparity is reduced to zero, resulting in zero error in the image space in both camera images simultaneously, the arm has reached its goal point in the task space—the only point at which the visual disparity equals zero. This work does not, however, control the end effector towards specific target orientations using visual feedback. The work of Corke *et. al.* [14] presents a decoupled approach to IBVS that solves one of the issues of direct image-space trajectories corresponding to contorted task-space trajectories by decoupling the z-axis components of the controller. This process then requires the arm to reach an intermediate position at a certain distance from the object before converging to its 3D location as opposed to the single motion produced by our system. The work of Garg *et. al.* [15] presents a stereo IBVS arm which reaches to a target specified by a laser pointer, but uses often noisy 3D point clouds to reach the object. The work of Ma *et. al.* [6] presents a 3-camera IBVS system with a single eye-in-hand camera and two eye-to-hand microscopic cameras, to grasp and align small size components. The eye-in-hand camera is used to align the end effector with the target object, and the two eye-to-hand cameras are used to ensure the grasped object's alignment with its target position for assembly by reaching the correct relative pose in both images, ensuring convergence to a 3D target point. Our system converges to 3D target positions as well, using only two cameras and no prior knowledge or reference image of the target, thus making it suitable for unstructured environments.

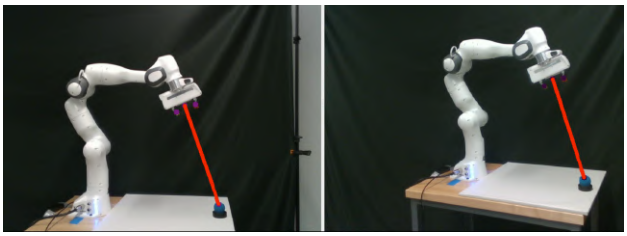


Fig. 2: Left and right camera image feature error vectors (red) between center point of fingers (purple) and target (ball)

We take inspiration from the work of Hager *et. al.* [12] by using IBVS that minimizes the error determined by the visual disparity between the two images to the target. Our work has real-time integration of control, using not only the camera poses in the image Jacobian calculation as in Maru *et. al.* [13], but also the robot forward kinematics for direct hand-eye mapping and feedback that can control the end-effector approach angle as well. The resulting controller produces output velocities that result in direct trajectories in simulation without decoupling the z-axis components of the controller.

The demonstrations in this paper use a simple point-and-click interface to provide grasping point locations to be used as goal references for the visual servoing algorithm. However, various automated grasping point detection algorithms can be coupled with our visual servoing scheme as well. For example, the work of Saxena *et. al.* [10] uses 2D images to identify good grasping points on a target object. This algorithm or similar algorithms could provide the 2D goal points for our visual servoing scheme to grasp the object in 3D space.

III. Methodology

A. Problem Statement

Given a stereo camera system with the end effector and target visible within the left and right images, our objective is to generate arm joint velocities to minimize the error vector detected between the end effector and target in the images (see Fig. 2). Once the target is reached in both images, the disparity between the images dictates that the target will be reached in the arm's task space as well.

B. Base Visual Servoing Scheme

To produce the joint velocities required to minimize the 2D image feature error vector, we use a variation of the visual servoing controller as shown in the "Tutorial on Visual Servo Control" by Hutchinson *et. al.* [1]. We begin by defining the 2D image feature error vector for both cameras in Eq. 1

$$\mathbf{e} = \mathbf{g}^* - \mathbf{p}^* \tag{1}$$

in which $\mathbf{e}$ represents our 2D feature error vector, a 4x1 vector with the features in the left camera image frame occupying the top two rows and the corresponding features in the right camera image frame occupying the bottom two rows. The same left and right camera image frame format is followed for $\mathbf{g}^*$ and $\mathbf{p}^*$ (4x1 vectors), which represent the observed 2D image features of the target position and the gripper position, respectively. The gripper position, $\mathbf{p}^*$, represents the observed center point calculated between the two fingers of the robot gripper.

$\mathbf{p}^*$ can be further defined as a function of the end effector position with respect to the camera frame, $\begin{bmatrix} p_x, p_y, p_z \end{bmatrix}^\mathbf{T}$, using the camera projection function as shown in Eq. 2, where f represents the camera focal length.

$$\mathbf{p}^* := -f \begin{bmatrix} p_x/p_z \\ p_y/p_z \end{bmatrix} \tag{2}$$

We obtain our feature velocity in image space, $\mathbf{v}$, in Eq. 3 by combining a control gain λ, the pseudoinverse of our image Jacobian, J_i^+, and our error vector from Eq. 1 [1].

$$\mathbf{v} = -\lambda J_i^+ \mathbf{e} \tag{3}$$

Our image Jacobian is further defined in Eq. 4, as $\mathbf{e}$ can be represented as a function of $\mathbf{p}$ to map the hand-eye coordination between observations in image space and positions in task space.

$$J_i^+ = \frac{\partial \mathbf{e}}{\partial \mathbf{p}} \tag{4}$$

We calculate the joint velocities required to minimize the 2D image feature error vector by combining the pseudoinverse of our robot manipulator Jacobian, J_r^+, with our feature velocity, $\mathbf{v}$, as shown in Eq. 5.

$$\dot{\mathbf{q}} = J_r^+ \mathbf{v} \tag{5}$$

Our robot manipulator Jacobian is further defined in Eq. 6, as $\mathbf{p}$ is a function of $\mathbf{q}$, mapping from task space to joint space.

$$J_r^+ = \frac{\partial \mathbf{p}}{\partial \mathbf{q}} \tag{6}$$

Now we substitute our feature velocity $\mathbf{v}$ from Eq. 3 into Eq. 5, to map from image space to joint space in Eq. 7.

$$\dot{\mathbf{q}} = -\lambda J_r^+ J_i^+ \mathbf{e} \tag{7}$$

We then combine the two Jacobians into one, J_c in Eq. 8, following the general equation of a visual servoing scheme.

$$\dot{\mathbf{q}} = -\lambda J_c^+ \mathbf{e} \tag{8}$$

C. Orientation Visual Servoing Scheme

To reach desired approach angles using only 2D image features for feedback, we modify the image feature error vector of our base visual servoing scheme from Eq. 1. We add a third entry representing the orientation relative to the positive x-axis of the image space to each of the left and right feature error vector components, providing orientation control along the axis perpendicular to each camera's image plane. We control orientation only about this axis to maintain a clear view of the fingers of the gripper and more accurately calculate the 2D orientation. This orientation is defined by obtaining the polar coordinates angle of a line between the center point of the two fingers, $\mathbf{p}^*$ from Eq. 2, and the pixel coordinates of the finger facing the positive x-axis of the image plane, (x_f, y_f). This formulation is shown in Eq. 9.

$$\mathbf{p}^* := \begin{bmatrix} -fp_x/p_z \\ -fp_y/p_z \\ \text{atan2}(y_f + fp_y/p_z, x_f + fp_x/p_z) \end{bmatrix} \tag{9}$$

Meanwhile, the desired approach angle of the target position with respect to the positive x-axis of the image plane is specified when defining the target position by setting the angle. With this new 6x1 feature error vector, we can simply continue the remainder of the derivation in Section III-B to produce a control scheme that minimizes error with respect to desired approach angles as well.

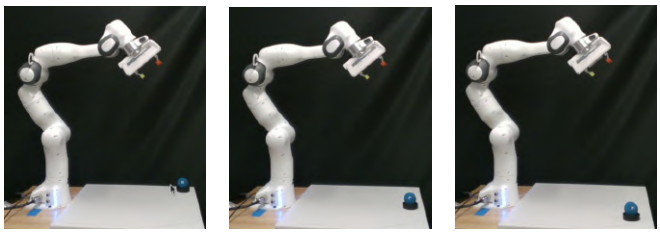
IV. Experiment Results and Discussion

A. Robot Setup

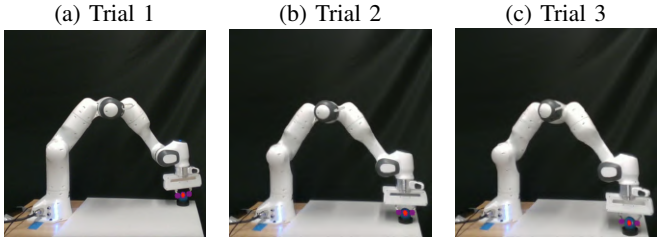
We present a detailed evaluation of our control scheme using the 7DOF Franka Emika Panda robot arm, as well as several proof of concept experiments with the Boston Dynamics Atlas Humanoid Robot. As the Panda arm does not have a built-in eye-to-hand stereo camera setup, we create one by setting up two RealSense cameras on tripods pointed towards the robot. The cameras are each set up roughly 1.7 m away from the robot, with a baseline of roughly 0.3 m apart from each other, to ensure an adequate disparity between the left and right camera images and full view of both fingers of the gripper as shown in Fig. 2. Despite the RealSense camera having a depth camera and two stereo IR cameras, we use only the RGB camera on each RealSense. The cameras' pose with respect to the Panda arm's base frame is calibrated using Nvidia's DREAM framework [16].

B. Grasping a ball at various depths away from the cameras

To show our system's ability to grasp objects in 3D space using only 2D image feedback, we set up a racquetball on a stand within the workspace of the Panda arm in three different target locations. As seen in Fig. 3, the ball is placed at various distances away from the cameras. The cameras' poses and the starting configuration of the arm are consistent for each trial. To select the 2D goal position, we click on the corresponding reference point in each of the left and right camera images (facilitated by a small tape marker) and do not specify a goal approach angle.



(a) Trial 1 (b) Trial 2 (c) Trial 3



(d) Trial 1 (e) Trial 2 (f) Trial 3

Fig. 3: Grasping a ball at varying depths from the cameras

The arm successfully grasps the racquetball in each configuration, as shown by the feature error vector over time shown in the graphs in Fig. 4. Detailed information about the control performance is provided in Table I. Thus, we have shown our stereo visual servoing control scheme can reach locations in 3D space using only 2D images for feedback.

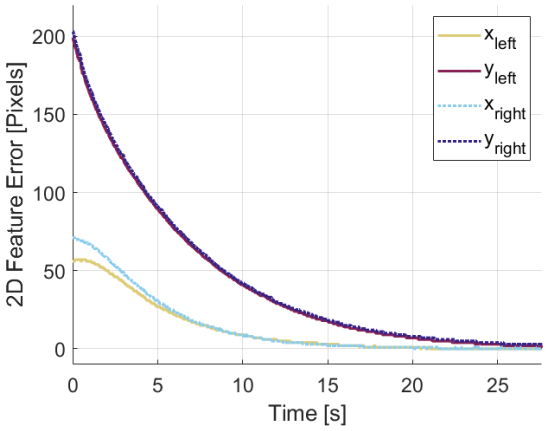


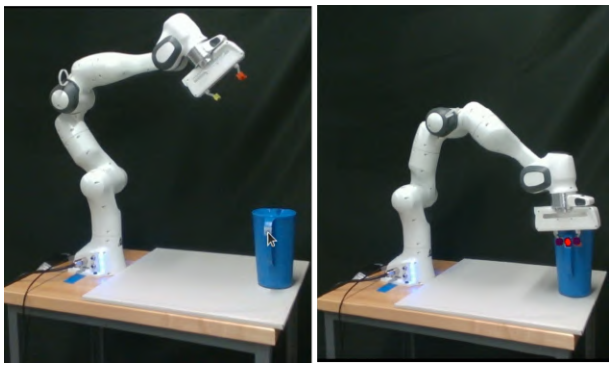
Fig. 4: Feature error over time for grasping ball in Trial 3

TABLE I: Grasping a Ball Steady State Error (SSE)

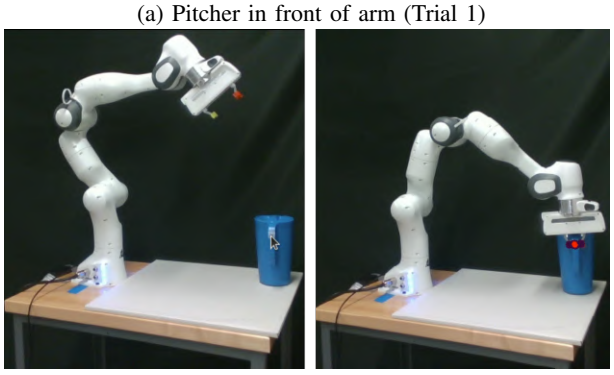
Trial	Left Image SSE (pixels)	Right Image SSE (pixels)	Left Image % Error	Right Image % Error
1	2.0	2.0	1.0%	0.9%
2	3.0	0.0	1.3%	0.0%
3	2.0	2.0	0.8%	0.8%

C. Picking Up a Pitcher Using Point-and-Click

To show our system's ability to enable the grasping of a practical object, we place a pitcher at two different positions in the Panda arm's workspace. As shown in Fig. 5, we click on a corresponding point in the left and right camera images



(a) Pitcher in front of arm (Trial 1)



(b) Pitcher to left of arm (Trial 2)

Fig. 5: Grasping a pitcher handle using point-and-click

on the handle of the pitcher (facilitated by a small tape marker).

The feature error over time graphs in Fig. 6 show similar performance in grasping the pitcher as in the racquetball experiments. Table II shows that our system can complete the grasp while maintaining a low steady state error at maximum of 1.2% of starting error.

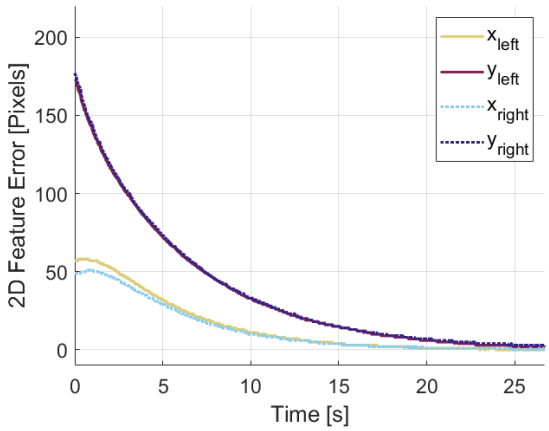


Fig. 6: Feature error over time to grasp pitcher in Trial 1

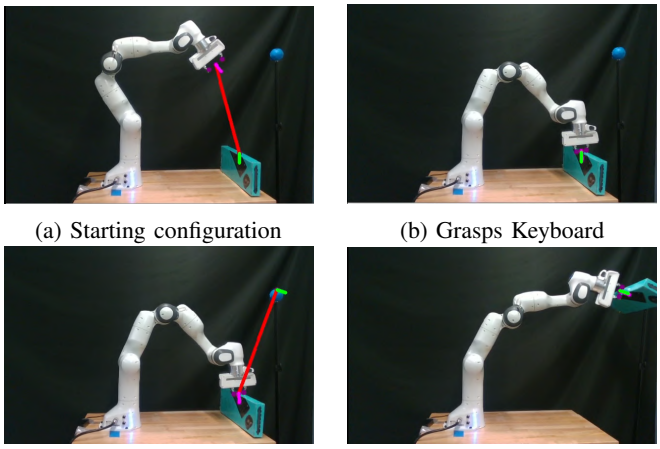
D. Pick and Place with Orientation-Inclusive Feature Vector

To show our system's ability to control for the orientation of the end effector using only 2D image data for feedback, we set up an experiment in which the gripper must grasp a

TABLE II: Grasping a Pitcher Steady State Error (SSE)

Trial	Left Image SSE (pixels)	Right Image SSE (pixels)	Left Image % Error	Right Image % Error
1	2.0	2.0	1.1%	1.1%
2	2.0	2.0	1.2%	1.1%

vertically-standing keyboard box.To successfully grasp the object, the gripper must have a vertical approach angle. Then, to place the object the gripper must have a horizontal approach angle. We specify each of these target angles in image space by entering a desired approach angle in the terminal window after clicking the target points. For this experiment, the first target point and approach angle are clicked and specified, and then an additional point and orientation are added to the queue. Once the box is grasped, the second target point is loaded from the queue and the arm immediately servos to the placing position and angle for the keyboard.



(a) Starting configuration (b) Grasps Keyboard

(c) Switching to place target (d) Places keyboard

Fig. 7: Pick and place a keyboard in a specific orientation

To complete the grasp, we set the desired orientation to 90 degrees such that the gripper will reach a vertical grasping configuration. As seen in Fig. 8 and Tables III and IV, our system reaches the pick and place target within 1.6% of the starting error in both pixel location and desired orientation.

TABLE III: Pick and Place Position Steady State Error (SSE)

Trial Half	Left Image SSE (pixels)	Right Image SSE (pixels)	Left Image % Error	Right Image % Error
Pick	3.0	2.0	1.6%	1.0%
Place	2.2	1.0	1.0%	0.4%

E. Atlas Humanoid Robot Experiments

We also evaluated our system on the physical Atlas robot. The robot has two 7-DOF arms with Robotiq 3-Finger Adaptive Robot Grippers, of which we use the right arm as the manipulator arm for our testing. We set the robot's starting arm positions and execute walking by using TOUGH

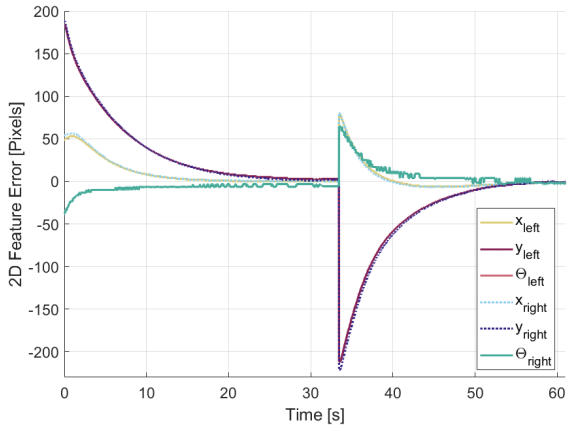


Fig. 8: Feature error over time to pick and place the keyboard

TABLE IV: Pick and Place Orientation Steady State Error (SSE)

Trial Half	Left Image SSE (degrees)	Right Image SSE (degrees)	Left Image % Error	Right Image % Error
Pick	5.7	0.0	0.2%	0.0%
Place	3.3	1.8	1.0%	0.0%

APIs, by Jagtap *et. al.* [17]. The output accelerations from our low-level controller are sent to Atlas's IHMC controllers by Koolen *et. al.* [18] to directly control the arm. The IHMC controllers automatically maintain the robot's balance if arm motion shifts Atlas's zero moment point outside the support polygon. Atlas's IHMC controllers also supply the actual joint angles and velocities, which we filter using a rolling average to use as feedback to our controller.

For perception, the robot uses a MultiSense SL 3D sense head, which includes a LiDAR and RGB-D cameras that we only access the 2D image data from. The MultiSense SL only detects RGB images in the left camera, so we use the grayscale images from both the left and right cameras, which are published at 30Hz. Thus, we use the KCF tracker of Henriques *et. al.* [19] to track the gripper fingertips and target object. To ensure high contrast with the background, we wrap the fingertips in white tape with a black X drawn on to provide a consistent feature reference when selecting points by clicking on the fingers in the image using OpenCV's Region of Interest (ROI) selector. The ball also has a black X drawn on it for the same reason. Consistent feature references ensure the disparity between the stereo cameras is properly reflected in the tracked features.

We start Atlas at about 1.5 meters away from the target and have Atlas walk five steps forward such that the target is now within Atlas's workspace. Then, we select the target object in the camera image and execute our visual servoing controller pipeline to grasp the target from this new position. The graph of feature error over time is shown in Fig. 9 and control performance is presented in Table V. We would like to note that when the arm reaches for the target, the Atlas robot's body does slight autonomous balancing motions

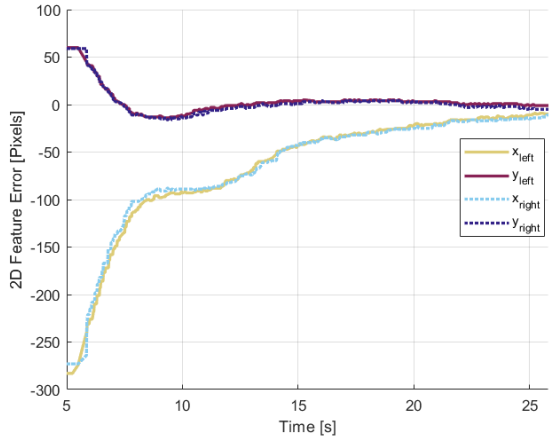
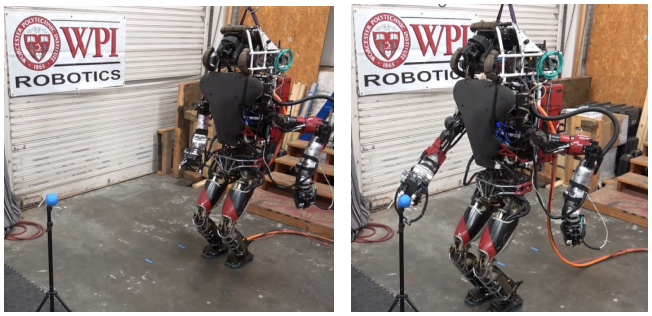


Fig. 9: Image plane error (in pixels) in left and right cameras to reach a target after walking to a nearby position

TABLE V: Walking then Grasping Steady State Error (SSE)

Left Image SSE (pixels)	Right Image SSE (pixels)	Left Image % Error	Right Image % Error
9.91 ± 0.64	13.84 ± 1.28	3.5%	4.3%

since the center of mass of the system continuously changes. Still, the visual servoing algorithm achieves convergence with satisfactory control performance.



(a) Trial 1 (b) Trial 2

Fig. 10: Grasping a ball from an uncertain starting configuration after walking

V. Conclusions and Future Work

The results of our experiments show that our algorithm can successfully grasp targets, including at desired pitch approach angles using stereo IBVS without stereo reconstruction. With the Panda robot, we show that the algorithm can reach points in 3D space at varying depths from the cameras, using only 2D images for feedback. We additionally show that our system can pick up a practical object by its handle and can pick and place a keyboard box at specified approach angles. Future work would explore the proper placement of an additional pair of cameras to control the full 6DOF pose of the gripper using only RGB images to overcome the limitation of controlling only the pitch angle.

Fig. 11: *Left:* Left error vector (red) in left camera image calculated between features tracked in the magenta boxes, *Right:* Right error vector (red) shown in right camera image calculated between features tracked in the yellow boxes

We show that the algorithm can achieve convergence even from uncertain starting configurations in our walking then grasping experiment. We also show that this system is modular enough to be implemented on a variety of robots with minimal changes, as the same control scheme is used for both the Panda and Atlas robot with only minor implementation-specific changes required.

One of the next steps for this work is to move towards vision-based full-body control, thus extending the workspace of the robot. By implementing this controller on Atlas, we have the potential to extend this work to moving the legs and torso of Atlas to reach targets outside its workspace in the starting standing position. We would like to thank WARNER, WPI's Atlas robot for its service in completing its final experiments with us before its retirement.

This work could be integrated with a 2D image-based grasp synthesis algorithm like the work of Saxena *et. al.* [10] to autonomously generate the target in the images and then our algorithm could minimize the error in the image space to reach the grasping position and complete the grasp.

This system presents a novel approach to stereo image-based visual servoing that requires no stereo reconstruction in order to reach targets, including at specified approach angles, moving towards robust control of vision-based robots with minimal data and execution steps required for convergence.

References

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.

[2] S. Noh, C. Park, and J. Park, "Position-based visual servoing of multiple robotic manipulators: verification in gazebo simulator," in *2020 international conference on information and communication technology convergence (ICTC)*. IEEE, 2020, pp. 843–846.

[3] R. S. Sharma, S. Shukla, L. Behera, and V. K. Subramanian, "Position-based visual servoing of a mobile robot with an automatic extrinsic calibration scheme," *Robotica*, vol. 38, no. 5, pp. 831–844, 2020.

[4] P. Yu, N. Tan, and M. Mao, "Position-based visual servo control of dual robotic arms with unknown kinematic models: A cerebellum-inspired approach," *IEEE/ASME Transactions on Mechatronics*, 2023.

[5] J. Haviland, F. Dayoub, and P. Corke, "Control of the final-phase of closed-loop visual grasping using image-based visual servoing," *arXiv preprint arXiv:2001.05650*, 2020.

[6] Y. Ma, X. Liu, J. Zhang, D. Xu, D. Zhang, and W. Wu, "Robotic grasping and alignment for small size components assembly based on visual servoing," *The International Journal of Advanced Manufacturing Technology*, vol. 106, pp. 4827–4843, 2020.

[7] M. Costanzo, G. De Maria, C. Natale, and A. Russo, "Modeling and control of sampled-data image-based visual servoing with three-dimensional features," *IEEE Transactions on Control Systems Technology*, 2023.

[8] A. Saxena, H. Pandya, G. Kumar, A. Gaud, and K. M. Krishna, "Exploring convolutional networks for end-to-end visual servoing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3817–3823.

[9] E. G. Ribeiro, R. de Queiroz Mendes, and V. Grassi Jr, "Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation," *Robotics and Autonomous Systems*, vol. 139, p. 103757, 2021.

[10] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[11] C. Chen, H. Zhu, L. Wang, and Y. Liu, "A stereo visual-inertial slam approach for indoor mobile robots in unknown environments without occlusions," *IEEE Access*, vol. 7, pp. 185 408–185 421, 2019.

[12] G. D. Hager, W.-C. Chang, and A. S. Morse, "Robot hand-eye coordination based on stereo vision," *IEEE Control Systems Magazine*, vol. 15, no. 1, pp. 30–39, 1995.

[13] N. Maru, H. Kase, S. Yamada, A. Nishikawa, and F. Miyazaki, "Manipulator control by using servoing with the stereo vision," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 3. IEEE, 1993, pp. 1866–1870.

[14] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 507–515, 2001.

[15] V. Garg, T. Sharma, A. Kumar, and V. Rastogi, "Handaid: A seven dof semi-autonomous robotic manipulator," in *2020 5th International Conference on Control and Robotics Engineering (ICCRE)*. IEEE, 2020, pp. 37–41.

[16] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-robot pose estimation from a single image," in *International Conference on Robotics and Automation (ICRA)*, 2020. [Online]. Available: https://arxiv.org/abs/1911.09231

[17] V. Jagtap, S. Agarwal, A. Wagh, and M. Gennert, "Transportable open-source application program interface and user interface for generic humanoids: Tough," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, p. 1729881420921607, 2020.

[18] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot atlas," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1650007, 2016.

[19] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 583–596, 2014.