

Video Object Segmentation-based Visual Servo Control and Object Depth Estimation on a Mobile Robot

Brent A. Griffin Victoria Florence Jason J. Corso
University of Michigan
{griffb,vflorenc,jjcorso}@umich.edu

Abstract

To be useful in everyday environments, robots must be able to identify and locate real-world objects. In recent years, video object segmentation has made significant progress on densely separating such objects from background in real and challenging videos. Building off of this progress, this paper addresses the problem of identifying generic objects and locating them in 3D using a mobile robot with an RGB camera. We achieve this by, first, introducing a video object segmentation-based approach to visual servo control and active perception and, second, developing a new Hadamard-Broyden update formulation. Our segmentation-based methods are simple but effective, and our update formulation lets a robot quickly learn the relationship between actuators and visual features without any camera calibration. We validate our approach in experiments by learning a variety of actuator-camera configurations on a mobile HSR robot, which subsequently identifies, locates, and grasps objects from the YCB dataset and tracks people and other dynamic articulated objects in real-time.

1. Introduction

Visual servo control (VS), using visual data in the servo loop to control a robot, is a well-established field [11, 28]. Using features from RGB images, VS has been used for positioning UAVs [26, 43] and wheeled robots [35, 42], manipulating objects [29, 54], and even laparoscopic surgery [56]. While this prior work attests to applicability of VS, generating robust visual features for VS in unstructured environments with generic objects (e.g., without fiducial markers) remains an open problem.

On the other hand, video object segmentation (VOS), the dense separation of objects in video from background, has made recent progress on real, unstructured videos. This progress is due in part to the introduction of multiple benchmark datasets [47, 49, 57], which evaluate VOS methods across many challenging categories, including moving cam-



Figure 1. **RGBD View of Cluttered Scene.** Using an RGB image (top left), HSR identifies and segments five target objects (top right). However, the associated depth image is unreliable (bottom left) and provides depth data for only one target (bottom right).

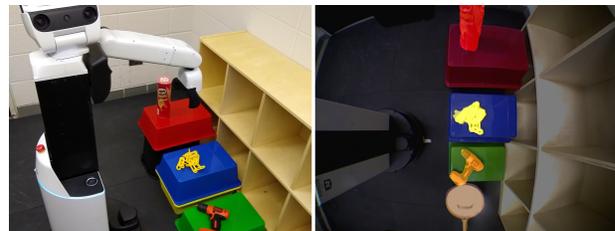


Figure 2. **Finding Objects in RGB.** With our approach, HSR segments, locates, and grasps objects using a single RGB camera.

eras, occlusions, objects leaving view, scale variation, appearance change, edge ambiguity, multiple interacting objects, and dynamic background; these challenges frequently occur simultaneously. However, despite all of VOS's contributions to video understanding, we are unaware of any work that utilizes VOS for control.

To this end, this paper develops a VOS-based framework to address the problem of visual servo control in unstructured environments. We also use VOS to estimate depth without a 3D sensor (e.g., an RGBD camera in Figure 1 and

[20, 53]). Developing VOS-based features for control and depth estimation has many advantages. First, VOS methods are robust across a variety of unstructured objects and backgrounds, making our framework general to many settings. Second, many VOS methods operate on streaming images, making them ideal for tracking objects from a moving robot. Third, ongoing work in active and interactive perception enables robots to automatically generate object-specific training data for VOS methods [6, 21, 32, 41]. Finally, VOS remains a hotly studied area of video understanding, and future improvements in the accuracy and robustness of state-of-the-art segmentation methods will similarly improve our method.

The primary contribution of our paper is the development and experimental evaluation of video object segmentation-based visual servo control (VOS-VS). We demonstrate the utility of VOS-VS on a mobile robot equipped with an RGB camera to identify and position itself relative to many challenging objects from HSR challenges and the YCB object dataset [9]. To the best of our knowledge, this work is first use of video object segmentation for control.

A second contribution is our new Hadamard-Broyden update formulation, which outperforms the original Broyden update in experiments and enables a robot to learn the relationship between actuators and VOS-VS features online without any camera calibration. Using our update, our robot learns to servo with seven unique configurations across seven actuators and two cameras. To the best of our knowledge, this work is the first use of a Broyden update to directly estimate the *pseudoinverse* feature Jacobian for visual servo control on a robot.

A final contribution is introducing two more VOS-based methods, VOS-DE and VOS-Grasp. VOS-DE combines segmentation features with Galileo’s Square-cube law and active perception to estimate an object’s depth, which, with VOS-VS, provides an object’s 3D location. VOS-Grasp uses segmentation features for grasping and grasp-error detection. Thus, using our approach, robots can find and grasp objects using a single RGB camera (see Figure 2).

We provide source code and annotated YCB object training data at <https://github.com/griffbr/VOSVS>.

2. Related Work

2.1. Video Object Segmentation

Video object segmentation methods can be categorized as unsupervised, which usually rely on object motion [19, 24, 33, 46, 55], or semi-supervised, which segment objects specified in user-annotated examples [5, 13, 23, 36, 45, 60]. Of particular interest to the current work, semi-supervised methods learn the visual characteristics of a target object, which enables them to reliably segment dynamic *or* static objects. To generate our VOS-based features, we segment

objects using One-Shot Video Object Segmentation (OS-VOS) [8], which is state-of-the-art in VOS and has influenced other leading semi-supervised methods [40, 51].

2.2. Visual Servo Control

In addition to the visual servo literature cited in Section 1, this paper builds off of other methods for control design and feature selection. For control design, a technique using a hybrid input of 3D Cartesian space and 2D image space is developed in [39], with depth estimation provided externally. As a step toward more natural image features, Canny edge detection-based planar contours of objects are used in [14]. When designing features, work in [38] shows that z -axis features should scale proportional to the optical depth of observed targets. Finally, work in [15] controls z -axis motions using the longest line connecting two feature points for rotation and the square root of the collective feature-point-polygon area for depth; this approach addresses the Chaumette Conundrum presented in [10] but also requires that all feature points remain in the image. Notably, early VS methods require structured visual features (e.g., fiducial markers), while recent learning-based methods require manipulators with a fixed workspace [1, 30, 62].

Taking advantage of recent progress in computer vision, this paper introduces robust segmentation-based image features for visual servoing that are generated from ordinary, real-world objects. Furthermore, our features are rotation invariant, work when parts of an object are out of view or occluded, and do not require any particular object viewpoint or marking, making this work applicable to articulated and deformable objects (e.g., the yellow chain in Figures 1-2). Finally, our method enables visual servo control on a mobile manipulation platform, on which we also use segmentation-based features for depth estimation and grasping.

2.3. Active Perception

A critical asset for robot perception is taking actions to improve sensing and understanding of the environment, i.e., Active Perception (AP) [3, 4]. Compared to structure from motion [2, 31, 34], which requires feature matching or scene flow to relate images, AP exploits knowledge of a robot’s relative position to relate images and improve 3D reconstruction. Furthermore, AP methods select new view locations explicitly to improve perception performance [17, 50, 61]. In this work, we use active perception with VOS-based features to estimate an object’s depth. We complete our estimate during our robot’s approach to an object, and, by tracking the estimate’s convergence, we can collect more data if necessary. Essentially, by using an RGB camera and kinematic information that is already available, we estimate the 3D position of objects without any 3D sensors, including: LIDAR, which is cost prohibitive and color blind; RGBD cameras, which do not work in ambient sun-

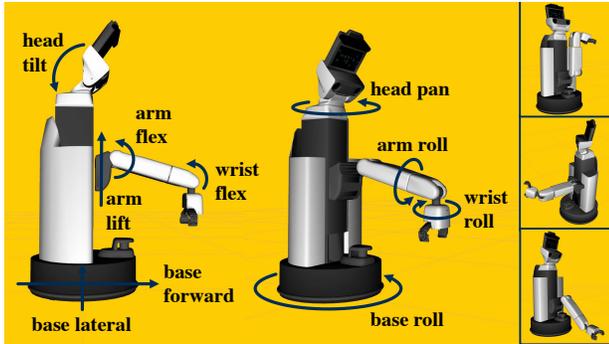


Figure 3. HSR Control Model.

light among other conditions (see Figure 1); and stereo cameras, which require calibration and feature matching. Even when 3D sensors are available, RGB-based methods provide an indispensable backup for perception [44, 52].

3. Robot Model and Perception Hardware

For our robot experiments, we use a Toyota Human Support Robot (HSR), which has a 4-DOF manipulator arm mounted on a torso with prismatic and revolute joints and a differential drive base [58, 59]. Using the revolute joint atop its differential drive base, we effectively control HSR as an omnidirectional robot. For visual servo control, we use the actuators shown in Figure 3 as the joint space $\mathbf{q} \in \mathbb{R}^{10}$,

$$\mathbf{q} = [q_{\text{head tilt}}, q_{\text{head pan}}, \dots, q_{\text{base roll}}]^T. \quad (1)$$

In addition to \mathbf{q} , HSR's end effector has a parallel gripper with series elastic fingertips for grasping objects; the fingertips have 135 mm maximum width.

For perception, we use HSR's base-mounted UST-20LX 2D scanning laser for obstacle avoidance and the head-mounted Xtion PRO LIVE RGBD camera and end effector-mounted wide-angle grasp camera for segmentation. The head tilt and pan joints act as a 2-DOF gimbal for the head camera, and the grasp camera moves with the arm and wrist joints; both cameras stream 640×480 RGB images.

A significant component of HSR's manipulation DOF comes from its mobile base. While many planning algorithms work well on high DOF arms with a stationary base, the odometer errors of HSR compound during trajectory execution and cause missed grasps. Thus, VS is well-suited for HSR and other mobile robots, providing visual feedback on an object's relative position during mobile manipulation.

4. Segmentation-based Visual Servo Control

4.1. Segmentation-based Features

Assume we are given an RGB image I containing an object of interest. Using VOS, we generate a binary mask

$$M = \text{vos}(I, \mathbf{W}), \quad (2)$$

where M consists of pixel-level labels $\ell_p \in \{0, 1\}$, $\ell_p = 1$ indicates pixel p corresponds to the segmented object, and \mathbf{W} are learned VOS parameters (details in Section 7.2).

Using M , we define the following VOS-based features

$$s_A := \sum_{\ell_p \in M} \ell_p \quad (3)$$

$$s_x := \frac{\sum_{\ell_p \in M, \ell_p=1} p_x}{s_A} \quad (4)$$

$$s_y := \frac{\sum_{\ell_p \in M, \ell_p=1} p_y}{s_A}, \quad (5)$$

where s_A is a measure of segmentation area by the number of labeled pixels, s_x is the x -centroid of the segmented object using x -axis label positions p_x , and s_y is the equivalent y -centroid. In addition to (3)-(5), we introduce more VOS features for depth estimation and grasping in Sections 5-6.

4.2. Visual Servo Control

Using VOS-based features for our visual servo control scheme, we define image feature error

$$\mathbf{e} := \mathbf{s}(I, \mathbf{W}) - \mathbf{s}^*, \quad (6)$$

where $\mathbf{s} \in \mathbb{R}^k$ is the vector of visual features found in image I using learned VOS parameters \mathbf{W} and $\mathbf{s}^* \in \mathbb{R}^k$ is the vector of desired feature values. In contrast to many VS control schemes, \mathbf{e} in (6) has no dependence on time, previous observations, or additional system parameters (e.g., camera parameters or 3D object models).

Typical VS approaches relate camera motion to \mathbf{s} using

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_c, \quad (7)$$

where $\mathbf{L}_s \in \mathbb{R}^{k \times 6}$ is a feature Jacobian relating the three linear and three angular camera velocities $\mathbf{v}_c \in \mathbb{R}^6$ to $\dot{\mathbf{s}}$. From (6)-(7), assuming $\dot{\mathbf{s}}^* = 0 \implies \dot{\mathbf{e}} = \dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_c$, we find the VS control velocities \mathbf{v}_c to minimize \mathbf{e} as

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}}_s^+ \mathbf{e}, \quad (8)$$

where $\widehat{\mathbf{L}}_s^+$ is the estimated pseudoinverse of \mathbf{L}_s and λ ensures an exponential decoupled decrease of \mathbf{e} [11]. Notably, VS control using (8) requires continuous, six degree of freedom (DOF) control of camera velocity.

To make (8) more general for discrete motion planning and fewer required control inputs, we modify (7)-(8) to

$$\Delta \mathbf{s} = \mathbf{J}_s \Delta \mathbf{q} \quad (9)$$

$$\Delta \mathbf{q} = -\lambda \widehat{\mathbf{J}}_s^+ \mathbf{e}, \quad (10)$$

where $\Delta \mathbf{q}$ is the change of $\mathbf{q} \in \mathbb{R}^n$ actuated joints, $\mathbf{J}_s \in \mathbb{R}^{k \times n}$ is the feature Jacobian relating $\Delta \mathbf{q}$ to $\Delta \mathbf{s}$, and $\widehat{\mathbf{J}}_s^+$ is the estimated pseudoinverse of \mathbf{J}_s . We command $\Delta \mathbf{q}$ directly to the robot joint space as our VOS-VS controller to minimize \mathbf{e} and reach the desired feature values \mathbf{s}^* in (6).

4.3. Hadamard-Broyden Update Formulation

In real visual servo systems, it is impossible to know the exact feature Jacobian (\mathbf{J}_s) relating control actuators to image features [11]. Instead, some VS methods estimate \mathbf{J}_s directly from observations [12]; among these, a few use the Broyden update rule [27, 29, 48], which iteratively updates online. In contrast to previous VS work, Broyden’s original paper provides a formulation to estimate the *pseudoinverse* feature Jacobian ($\widehat{\mathbf{J}}_s^+$) [7, (4.5)]. However, we found it necessary to augment Broyden’s formulation with a logical matrix \mathbf{H} , and define our new Hadamard-Broyden update

$$\widehat{\mathbf{J}}_s^+_{t+1} := \widehat{\mathbf{J}}_s^+_{t} + \alpha \left(\frac{(\Delta \mathbf{q} - \widehat{\mathbf{J}}_s^+_{t} \Delta \mathbf{e}) \Delta \mathbf{q}^T \widehat{\mathbf{J}}_s^+_{t}}{\Delta \mathbf{q}^T \widehat{\mathbf{J}}_s^+_{t} \Delta \mathbf{e}} \right) \circ \mathbf{H}, \quad (11)$$

where α determines the update speed, $\Delta \mathbf{q} = \mathbf{q}_t - \mathbf{q}_{t-1}$ and $\Delta \mathbf{e} = \mathbf{e}_t - \mathbf{e}_{t-1}$ are the changes in joint space and feature errors since the last update, and $\mathbf{H} \in \mathbb{R}^{n \times k}$ is a logical matrix coupling actuators to image features. In experiments, we initialize (11) using $\alpha = 0.1$ and $\widehat{\mathbf{J}}_s^+_{t=0} = 0.001\mathbf{H}$.

The Hadamard product with \mathbf{H} prevents undesired coupling between certain actuator and image feature pairs. In practice, we find that using the original Broyden update results in unpredictable convergence and learning gains for actuator-image feature pairs that are, in fact, unrelated. Fortunately, we find that using \mathbf{H} in (11) enables real-time convergence without any calibration on the robot for all of the experiment configurations in Section 7.3.

4.4. VOS-VS Configurations

We learn seven unique VOS-VS configurations using our HB update. Using s_x (4) and s_y (5) in e (6), we define error

$$\mathbf{e}_{x,y} := \mathbf{s}_{x,y}(M(I, \mathbf{W})) - \mathbf{s}^* = \begin{bmatrix} s_x \\ s_y \end{bmatrix} - \mathbf{s}^*. \quad (12)$$

Using $\mathbf{e}_{x,y}$ and HSR joints \mathbf{q} (1), we choose $\widehat{\mathbf{J}}_s^+$ in (11) as

$$\widehat{\mathbf{J}}_s^+ \approx \frac{\partial \mathbf{q}}{\partial \mathbf{e}_{x,y}} = \frac{\partial \mathbf{q}}{\partial s_{x,y}} = \begin{bmatrix} \frac{\partial q_{\text{head tilt}}}{\partial s_x} & \frac{\partial q_{\text{head tilt}}}{\partial s_y} \\ \frac{\partial q_{\text{head pan}}}{\partial s_x} & \frac{\partial q_{\text{head pan}}}{\partial s_y} \\ \vdots & \vdots \\ \frac{\partial q_{\text{base roll}}}{\partial s_x} & \frac{\partial q_{\text{base roll}}}{\partial s_y} \end{bmatrix}, \quad (13)$$

where $\widehat{\mathbf{J}}_s^+ \in \mathbb{R}^{10 \times 2}$. Note that in our Hadamard-Broyden update (11), each element $\frac{\partial q_i}{\partial s_j}$ in $\widehat{\mathbf{J}}_s^+$ is multiplied by element $\mathbf{H}_{i,j}$ in the Hadamard product. Thus, we configure the logical coupling matrix \mathbf{H} by setting $\mathbf{H}_{i,j} = 1$ if coupling actuated joint q_i with image feature s_j is desired. Using our update formulation (11), we learn $\widehat{\mathbf{J}}_s^+$ on HSR for the seven \mathbf{H} configurations listed in Table 1 and provide experimental results for each configuration in Section 7.3.

Table 1. VOS-VS Hadamard-Broyden Update Configurations. $\widehat{\mathbf{J}}_s^+$ values are learned online using our HB update formulation (11), enabling HSR to automatically learn the relationship between actuators and visual features without any camera calibration.

H (11) Config.	Camera	Learned $\frac{\partial q_i}{\partial s_j}$ in $\widehat{\mathbf{J}}_s^+$ (13)			
		s_x		s_y	
\mathbf{H}_{head}	Head	$q_{\text{head pan}}$	0.00173	$q_{\text{head tilt}}$	0.00183
$\mathbf{H}_{\text{arm lift}}$	Grasp	$q_{\text{arm lift}}$	-0.00157	$q_{\text{arm roll}}$	0.00321
$\mathbf{H}_{\text{arm wrist}}$	Grasp	$q_{\text{wrist flex}}$	-0.00221	$q_{\text{arm roll}}$	0.00445
$\mathbf{H}_{\text{arm both}}$	Grasp	$q_{\text{arm lift}}$	-0.00036	$q_{\text{arm roll}}$	0.00328
\mathbf{H}_{base}	Grasp	$q_{\text{base forward}}$	-0.00179	$q_{\text{base lateral}}$	0.00173
$\mathbf{H}_{\text{base grasp}}$	Grasp	$q_{\text{base forward}}$	-0.00040	$q_{\text{base lateral}}$	0.00040

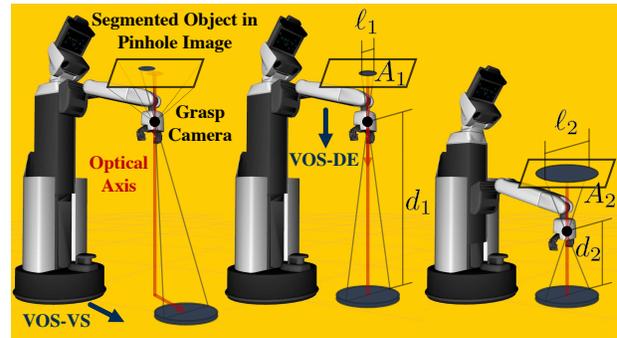


Figure 4. VOS-based Visual Servo and Depth Estimation. HSR first aligns an object with the camera’s optical axis then estimates the object’s depth as the camera approaches. Using Galileo’s Square-cube law (15), we estimate the object’s depth using changes in relative camera position and segmentation area.

5. Segmentation-based Depth Estimation

By combining VOS-based features with active perception, we are able to estimate the depth of segmented objects and approximate their 3D position. As shown in Figure 4, we initiate our depth estimation framework (VOS-DE) by centering the optical axis of our camera with a segmented object using the \mathbf{H}_{base} VOS-VS controller. This alignment minimizes lens distortion, which facilitates the use of an ideal camera model. Using the pinhole camera model [22], projections of objects onto the image plane scale inversely with their distance on the optical axis from the camera.

Thus, with the object centered on the optical axis, we can relate projection scale and object distance using

$$\ell_1 d_1 = \ell_2 d_2 \implies \frac{\ell_2}{\ell_1} = \frac{d_1}{d_2}, \quad (14)$$

where ℓ_1 is the projected length of an object measurement orthogonal to the optical axis, d_1 is the distance along the optical axis of the object away from the camera, and ℓ_2 is the projected measurement length at a new distance d_2 . Combining Galileo Galilei’s Square-cube law with (14),

$$A_2 = A_1 \left(\frac{\ell_2}{\ell_1} \right)^2 \implies A_2 = A_1 \left(\frac{d_1}{d_2} \right)^2, \quad (15)$$

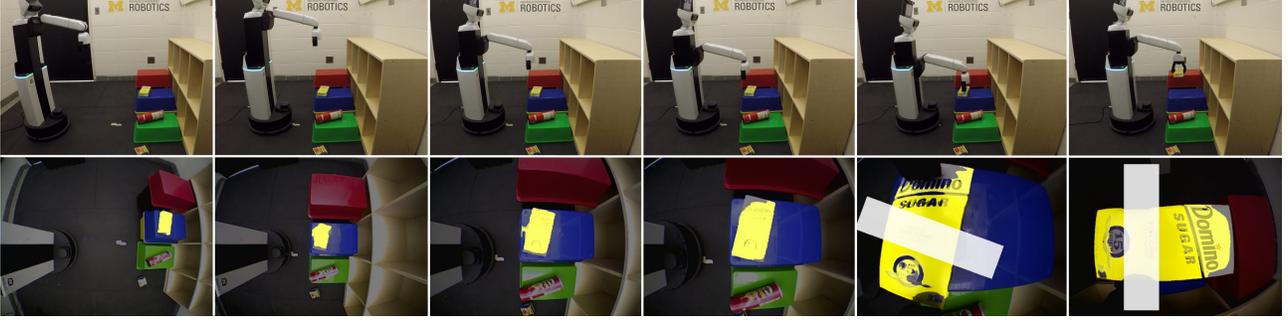


Figure 5. **VOS-based Grasping.** VOS-based visual servo control (columns 1 to 2), active depth estimation (2-4), and mobile robot grasping (5-6). Using our combined framework with a single RGB camera, HSR identifies the sugar box, locates it in 3D, and picks it up in real-time.

where A_1 is the projected object area corresponding to ℓ_1 and d_1 (see Figure 4). As the camera advances on the optical axis, we modify (15) to relate collected images using

$$d_1 \sqrt{A_1} = d_2 \sqrt{A_2} = c_{\text{object}}, \quad (16)$$

where c_{object} is a constant proportional to the orthogonal surface area of the segmented object. Also, using a coordinate frame with the z axis aligned with the optical axis,

$$d = z_{\text{camera}} - z_{\text{object}}, \quad (17)$$

where z_{camera} and z_{object} are the z -axis coordinates of the camera and object. Because the camera and object are both centered on the z axis, $x_{\text{camera}} = x_{\text{object}} = 0$ and $y_{\text{camera}} = y_{\text{object}} = 0$. Using (17) and s_A (3), we update (16) as

$$\begin{aligned} (z_{\text{camera},1} - z_{\text{object}}) \sqrt{s_{A,1}} &= (z_{\text{camera},2} - z_{\text{object}}) \sqrt{s_{A,2}} \\ &= c_{\text{object}}, \end{aligned} \quad (18)$$

where the object is assumed stationary between images (i.e., $\dot{z}_{\text{object}} = 0$) and the z_{camera} position is known from the robot's kinematics. Note that z_{camera} provides relative depth for VOS-DE and (18) identifies a key linear relationship between $\sqrt{s_A}$ and the distance between the object and camera.

Finally, after collecting a series of m measurements, we estimate the depth of the segmented object. From (18),

$$z_{\text{object}} \sqrt{s_{A,1}} + c_{\text{object}} = z_{\text{camera},1} \sqrt{s_{A,1}}, \quad (19)$$

which over the m measurements in $\mathbf{Ax} = \mathbf{b}$ form yields

$$\begin{bmatrix} \sqrt{s_{A,1}} & 1 \\ \sqrt{s_{A,2}} & 1 \\ \vdots & \vdots \\ \sqrt{s_{A,m}} & 1 \end{bmatrix} \begin{bmatrix} \hat{z}_{\text{object}} \\ \hat{c}_{\text{object}} \end{bmatrix} = \begin{bmatrix} z_{\text{camera},1} \sqrt{s_{A,1}} \\ z_{\text{camera},2} \sqrt{s_{A,2}} \\ \vdots \\ z_{\text{camera},m} \sqrt{s_{A,m}} \end{bmatrix}. \quad (20)$$

By solving (20) for \hat{z}_{object} and \hat{c}_{object} , we estimate the distance d in (17), and, thus, the 3D location of the object. In Section 7.4, we show that our combined VOS-VS and VOS-DE framework is sufficient for locating, approaching, and estimating the depth of a variety of unstructured objects.

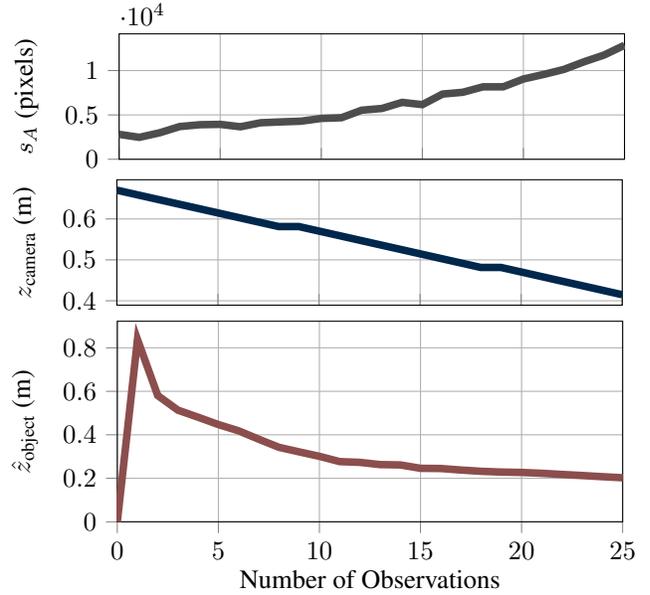


Figure 6. **Depth Estimate of Sugar Box.** Data collected and processed in real-time during the initial approach in Figure 5.

Remark: There are many methods to find approximate solutions to (20). In practice, we find that a least squares solution provides robustness to outliers caused by segmentation errors (see visual and quantitative example in Figures 5-6).

6. Segmentation-based Grasping

We develop a VOS-based method of grasping and grasp-error detection (VOS-Grasp). Assuming an object is centered and has estimated depth \hat{z}_{object} , we move z_{camera} to

$$z_{\text{camera, grasp}} = \hat{z}_{\text{object}} + z_{\text{gripper}}, \quad (21)$$

where z_{gripper} is the known z -axis offset between z_{camera} and the center of HSR's closed fingertips. Thus, when z_{camera} is at $z_{\text{camera, grasp}}$, HSR can reach the object at depth \hat{z}_{object} .

After moving to $z_{\text{camera, grasp}}$, we center the object directly underneath HSR's antipodal gripper using $\mathbf{H}_{\text{base grasp}}$ VOS-



Figure 7. **Experiment Objects from YCB Dataset.** Object categories are (from left to right) Food, Kitchen, Tool, and Shape. Spanning from 470 mm long to the 4 mm thick, we intentionally select many of the challenge objects to break our framework.

VS control. To find a suitable grasp location, we project and rotate a mask of the gripper, M_{grasp} , into the camera as shown in column 5 of Figure 5 and solve

$$\arg \min_{q_{\text{wrist roll}}} \mathcal{J}(q_{\text{wrist roll}}) = \frac{M \cap M_{\text{grasp}}(q_{\text{wrist roll}})}{M \cup M_{\text{grasp}}(q_{\text{wrist roll}})}, \quad (22)$$

where \mathcal{J} is the intersection over union (or Jaccard index [18]) of M_{grasp} and object segmentation mask M , and $M_{\text{grasp}}(q_{\text{wrist roll}})$ is the projection of M_{grasp} corresponding to HSR wrist rotation $q_{\text{wrist roll}}$. Thus, we grasp the object using the wrist rotation with least intersection between the object and the gripper, which is then less likely to collide with the object before achieving a parallel grasp.

After the object is grasped, we lift HSR’s arm to perform a visual grasp check. We consider a grasp complete if

$$s_{A,\text{raised}} > 0.5 s_{A,\text{grasp}}, \quad (23)$$

where $s_{A,\text{grasp}}$ is the object segmentation size s_A (3) during the initial grasp and $s_{A,\text{raised}}$ is the corresponding s_A after lifting the arm. If s_A decreases when lifting the arm, the object is further from the camera and not securely grasped. Thus, we quickly identify if a grasp is missed and regrasp as necessary. Note that this VOS-based grasp check can also work with other grasping methods [25, 37]. A complete demonstration of our VOS-based visual servo control, depth estimation, and grasping framework is shown in Figure 5.

7. ROBOT EXPERIMENTS

7.1. Experiment Objects

For most of our experiments, we use the objects from the YCB object dataset [9] shown in Figure 7. We use six objects from each of the food, kitchen, tool, and shape categories and purposefully choose some of the most difficult objects. To name only a few of the challenges for the selected objects: dimensions span from the 470 mm long pan to the 4 mm thick washer, most of the contours change with pose, and over a third of the objects exhibit specular reflection of overhead lights. To learn object recognition, we annotate ten training images of each object using HSR’s grasp camera with various object poses, backgrounds, and distances from the camera (see example image in Figure 2).

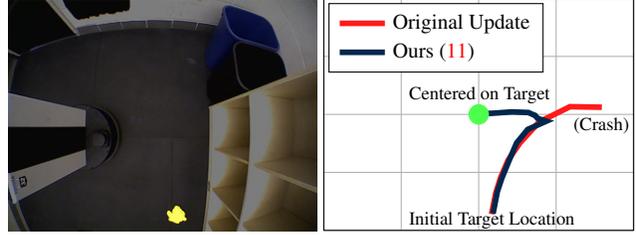


Figure 8. **Learning $\widehat{\mathbf{J}}_s^+$ for \mathbf{H}_{base} .** Visual servo trajectory of the target object in image space (right) using the original Broyden update (red) and our Hadamard-Broyden update (11) (blue). Starting with the same $\widehat{\mathbf{J}}_s^+_{t=0}$ and offset target location (yellow chain, left), the original update leads HSR into the wall while our update learns the correct visual servoing parameters to center HSR on the target.

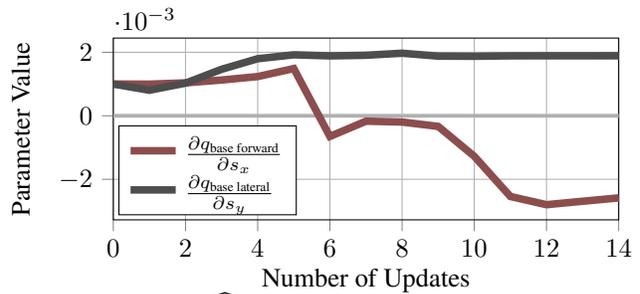


Figure 9. **Learning $\widehat{\mathbf{J}}_s^+$ Parameters for \mathbf{H}_{base} .** This plot corresponds to the fourteen Hadamard-Broyden updates used to learn visual servoing parameters in Figure 8. $\frac{\partial q_{\text{base forward}}}{\partial s_x}$ initializes with the incorrect sign but still converges using our update formulation.

7.2. Video Object Segmentation Method

We segment objects using OSVOS [8]. OSVOS uses a base network trained on ImageNet [16] to recognize image features, re-trains a parent network on DAVIS [47] to learn general video object segmentation, and then fine tunes for each of our experiment objects (i.e., each object has unique learned parameters \mathbf{W} in (2)). After learning \mathbf{W} , our VOS framework segments HSR’s 640×480 RGB images at 29.6 Hz using a single GPU (GTX 1080 Ti).

7.3. VOS-VS Results

Hadamard-Broyden Update We learn all of the VOS-VS configurations in Table 1 on HSR using the Hadamard-Broyden update formulation in (11). We initialize each configuration using $\widehat{\mathbf{J}}_s^+_{t=0} = 0.001 \mathbf{H}$, $\alpha = 0.1$, and a target object in view to elicit a step response from the VOS-VS controller (see Figure 8). Each configuration starts at a specific pose (e.g., \mathbf{H}_{base} uses the leftmost pose in Figures 4-5), and configurations use $s^* = [320, 240]'$ in (12), except for $\mathbf{H}_{\text{base grasp}}$, which uses $s^* = [220, 240]'$ to position grasps.

When initializing each configuration, after a few iterations of control inputs from (10) and updates from (11),

the learned $\widehat{\mathbf{J}}_s^+$ matrix generally shows convergence for any $\mathbf{H}_{i,j}$ component that is initialized with the correct sign (e.g., five updates for $\frac{\partial q_{\text{base lateral}}}{\partial s_y}$ in Figure 9). Components initialized with an incorrect sign generally require more updates to change directions and jump through zero during one of the discrete updates (e.g., $\frac{\partial q_{\text{base forward}}}{\partial s_x}$ in Figure 9). If an object goes out of view from an incorrectly signed component, we reset HSR’s pose and restart the update from the most recent $\widehat{\mathbf{J}}_s^+$. Once s^* is reached, the object can be moved to elicit a few more step responses for fine tuning. Table 1 shows the learned parameters for each configuration. In the remaining experiments, we set $\alpha = 0$ in (11) to reduce variability.

\mathbf{H}_{base} Results We show the step response of all $\widehat{\mathbf{J}}_s^+$ configurations in Table 1 by performing experiments centering the camera on objects placed at various viewpoints within each configuration’s starting pose. In Figure 10, both \mathbf{H}_{base} and $\mathbf{H}_{\text{base grasp}}$ exhibit a stable response. Our motivation to learn two base configurations is the increase in $s_{x,y}$ sensitivity to base motion as an object’s depth decreases. \mathbf{H}_{base} operates with the camera raised high above objects, while $\mathbf{H}_{\text{base grasp}}$ operates with the camera directly above objects to position for grasping. Thus, \mathbf{H}_{base} requires more movement than $\mathbf{H}_{\text{base grasp}}$ for the same changes in $s_{x,y}$. This difference is apparent in Table 1 from \mathbf{H}_{base} learning greater $\frac{\partial q_{\text{base}}}{\partial s}$ values and in Figure 10 from \mathbf{H}_{base} ’s smaller $s_{x,y}$ distribution for identical object distances.

\mathbf{H}_{arm} Results We show the step response of all arm-based VOS-VS configurations in Figure 11. Each configuration uses the same objects and starting pose. Although each configuration segments the pan and baseball, s^* is not reachable for these objects within any of the configured actuator spaces; $\mathbf{H}_{\text{arm wrist}}$ is the only configuration to center on all four of the other objects. The overactuated $\mathbf{H}_{\text{arm both}}$ has the most overshoot, while $\mathbf{H}_{\text{arm lift}}$ has the most limited range of camera positions but essentially deadbeat control.

\mathbf{H}_{head} Results Finally, we show the step response of \mathbf{H}_{head} in Figure 12. \mathbf{H}_{head} is the only configuration that uses HSR’s 2-DOF head gimbal and camera, and it exhibits a smooth step response over the entire image. Remarkably, even though \mathbf{H}_{head} uses the head’s camera, it still uses the same OSVOS parameters \mathbf{W} that are learned on grasp camera images; this further demonstrates the general applicability of VOS-VS in regards to needing no camera calibration.

7.4. Consecutive Mobile Robot Trials

We perform an experiment consisting of a consecutive set of mobile trials that simultaneously test VOS-VS and VOS-DE. Each trial consists of three unique YCB objects placed at different heights: one on the blue bin 0.25 m above the ground, one on the green bin 0.125 m above the ground, and one directly on the ground (see bin configura-

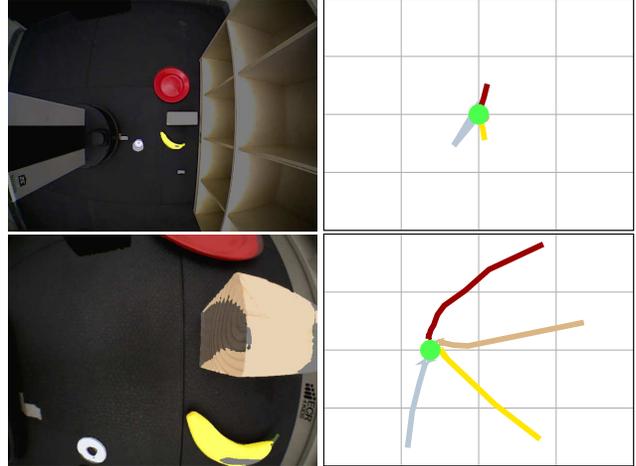


Figure 10. **Visual Servoing using Learned Parameters.** Initial view with segmented objects (left) and visual servo trajectories centering on each object (right). While objects are identically placed for the \mathbf{H}_{base} (top) and $\mathbf{H}_{\text{base grasp}}$ (bottom) experiments, each configuration has learned the correct scale of actuation to center on objects from its own visual perspective. Note that in the \mathbf{H}_{base} view, the wood block starts very close to s^* (green dot).

tion in Figure 2). The trial configurations and corresponding results are provided in Table 2. VOS-VS is considered a success (“X”) if HSR locates and centers on the object for depth estimation. VOS-DE is considered a success if HSR achieves $z_{\text{camera, grasp}}$ (21) such that HSR can close its grippers on the object without hitting the underlying surface and z_{camera} does not move past the top surface of the object.

Across all 24 objects, VOS-VS has a 83% success rate. VOS-DE, which is only applicable when VOS-VS succeeds, has a 50% success rate. By category, food objects have the highest success (100% VOS-VS, 83% VOS-DE) and kitchen objects have the lowest (50% VOS-VS, 66% VOS-DE). Failures are caused by segmentation errors. Although VOS-VS can center on a poorly segmented object, VOS-DE fails if there are erratic changes in segmentation area (we provide examples in the supplementary material). Additionally, VOS-DE’s margin for success varies between objects (e.g., the smallest margin is the 4 mm thick washer).

7.5. Additional Experiments

Pick-and-place Challenges We perform additional experiments for our VOS-based methods, including our work in the TRI-sponsored HSR challenges. These challenges consist of timed trials for pick-and-place tasks with randomly scattered, non-UCB objects (e.g., the banana peel in Figure 13). These challenges are a particularly good demonstration of VOS-VS and VOS-Grasp. We provide additional figures for these experiments in the supplementary material.

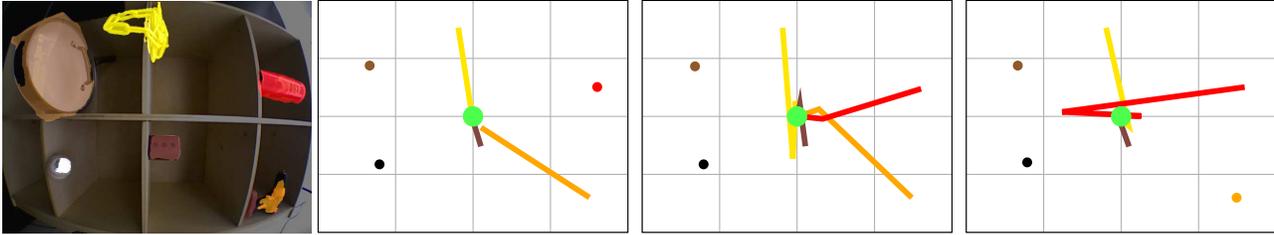


Figure 11. Initial view of objects and visual servo trajectories using $H_{\text{arm lift}}$ (center left), $H_{\text{arm wrist}}$ (center right), and $H_{\text{arm both}}$ (right).

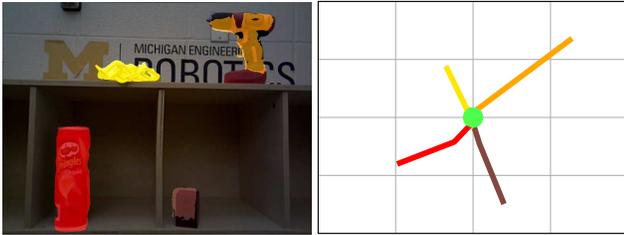


Figure 12. Initial view and visual servo trajectories using H_{head} .

Table 2. **Consecutive Mobile Robot Trial Results.** All results are from a single consecutive set of mobile HSR trials. Across all of the challenge objects, VOS-VS has a 83% success rate. Except for one VOS-DE trial, the food objects were a complete success.

Item	Object Category	Support Height (m)	Success	
			VS	DE
Chips Can	Food	0.25	X	X
Potted Meat	Food	0.125	X	X
Plastic Banana	Food	Ground	X	X
Box of Sugar	Food	0.25	X	X
Tuna	Food	0.125	X	
Gelatin	Food	Ground	X	X
Mug	Kitchen	0.25	X	X
Softscrub	Kitchen	0.125		N/A
Skillet with Lid	Kitchen	Ground		N/A
Plate	Kitchen	0.25	X	X
Spatula	Kitchen	0.125		N/A
Knife	Kitchen	Ground	X	
Power Drill	Tool	0.25	X	X
Marker	Tool	0.125	X	
Padlock	Tool	Ground	X	
Wood	Tool	0.25	X	
Spring Clamp	Tool	0.125	X	
Screwdriver	Tool	Ground	X	
Baseball	Shape	0.25	X	
Plastic Chain	Shape	0.125	X	
Washer	Shape	Ground	X	
Stacking Cup	Shape	0.25	X	X
Dice	Shape	0.125		N/A
Foam Brick	Shape	Ground	X	X

Dynamic Articulated Objects Finally, we perform additional VOS-VS experiments with dynamic articulated objects. Using H_{base} , HSR tracks a plastic chain across the room in real-time as we kick it and throw it in a variety of unstructured poses; we can even pick up the chain and use it to guide HSR’s movements from the grasp camera.

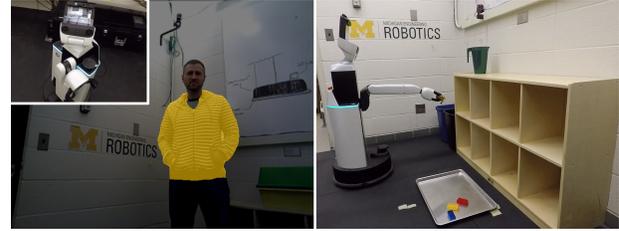


Figure 13. **Additional Experiments.** Using VOS-VS, HSR is able to track dynamic objects like people in real-time, making VOS-VS a useful tool for human-robot cooperation (left). HSR taking banana peel to garbage for a pick-and-place challenge (right).

In addition, by training OSVOS to recognize an article of clothing, HSR reliably tracks a person moving throughout the room using H_{head} (see Figure 13). Experiment videos are available at: <https://youtu.be/hlog5FV9RLs>.

8. Conclusions and Future Work

We develop a video object segmentation-based approach to visual servo control, depth estimation, and grasping. Visual servo control is a useful framework for controlling a physical robot system from RGB images, and video object segmentation has seen rampant advances within the computer vision community for densely segmenting unstructured objects in challenging videos. The success of our segmentation-based approach to visual servo control in mobile robot experiments with real-world objects is a tribute to both of these communities and the initiation of a bridge between them. Future developments in video object segmentation will improve the robustness of our method and, we expect, lead to other innovations in robotics.

A significant benefit of our segmentation-based framework is that it only requires an RGB camera combined with robot actuation. For future work, we are improving RGB-based depth estimation and grasping by comparing images collected from more robot poses, thereby leveraging more information and making our 3D understanding of the target object more complete.

Acknowledgment Toyota Research Institute (“TRI”) provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

- [1] P. Abolghasemi, A. Mazaheri, M. Shah, and L. Boloni. Pay attention! - robustifying a deep visuomotor policy through task-focused visual attention. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2
- [2] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images-a review. *Proceedings of the IEEE*, 76(8):917–935, Aug 1988. 2
- [3] R. Bajcsy. Active perception. *Proceedings of the IEEE (Invited Paper)*, 76(8):966–1005, Aug 1988. 2
- [4] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, Feb 2018. 2
- [5] L. Bao, B. Wu, and W. Liu. CNN in MRF: video object segmentation via inference in A cnn-based higher-order spatio-temporal MRF. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [6] B. Browatzki, V. Tikhanoff, G. Metta, H. H. Bulthoff, and C. Wallraven. Active in-hand object recognition on a humanoid robot. *IEEE Transactions on Robotics*, 30(5):1260–1269, Oct 2014. 2
- [7] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965. 4
- [8] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6
- [9] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics Automation Magazine*, 22(3):36–52, Sep. 2015. 2, 6
- [10] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. J. Kriegman, G. D. Hager, and A. S. Morse, editors, *The confluence of vision and control*, pages 66–78, London, 1998. Springer London. 2
- [11] F. Chaumette and S. Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics Automation Magazine*, 13(4):82–90, Dec 2006. 1, 3, 4
- [12] F. Chaumette and S. Hutchinson. Visual servo control. ii. advanced approaches [tutorial]. *IEEE Robotics Automation Magazine*, 14(1):109–118, March 2007. 4
- [13] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [14] G. Chesi, E. Malis, and R. Cipolla. Automatic segmentation and matching of planar contours for visual servoing. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 3, pages 2753–2758 vol.3, April 2000. 2
- [15] P. I. Corke and S. A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4):507–515, Aug 2001. 2
- [16] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6
- [17] R. Eidenberger and J. Scharinger. Active perception and scene modeling by planning with probabilistic 6d object poses. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1036–1043, Oct 2010. 2
- [18] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 6
- [19] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *British Machine Vision Conference (BMVC)*, 2014. 2
- [20] M. Ferguson and K. Law. A 2d-3d object detection system for updating building information models with mobile robots. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1357–1365, Jan 2019. 2
- [21] V. Florence, J. J. Corso, and B. Griffin. Self-supervised robot in-hand object learning. *CoRR*, abs/1904.00952, 2019. 2
- [22] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002. 4
- [23] B. A. Griffin and J. J. Corso. Bubblenets: Learning to select the guidance frame in video object segmentation by deep sorting frames. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2
- [24] B. A. Griffin and J. J. Corso. Tukey-inspired video object segmentation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 2
- [25] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt. High precision grasp pose detection in dense clutter. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 598–605, Oct 2016. 6
- [26] N. Guenard, T. Hamel, and R. Mahony. A practical visual servo control for an unmanned aerial vehicle. *IEEE Transactions on Robotics*, 24(2):331–340, April 2008. 1
- [27] K. Hosoda and M. Asada. Versatile visual servoing without knowledge of true jacobian. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 186–193 vol.1, Sep. 1994. 4
- [28] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, Oct 1996. 1
- [29] M. Jagersand, O. Fuentes, and R. Nelson. Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, volume 4, pages 2874–2880 vol.4, April 1997. 1, 4
- [30] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping

- via randomized-to-canonical adaptation networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2
- [31] Y. Kasten, M. Galun, and R. Basri. Resultant based incremental recovery of camera pose from pairwise matches. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1080–1088, Jan 2019. 2
- [32] M. Krainin, P. Henry, X. Ren, and D. Fox. Manipulator and object tracking for in-hand 3d object modeling. *The International Journal of Robotics Research*, 30(11):1311–1327, 2011. 2
- [33] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. 2
- [34] H. C. Longuet-Higgins. Readings in computer vision: Issues, problems, principles, and paradigms. chapter A Computer Algorithm for Reconstructing a Scene from Two Projections, pages 61–62. 1987. 2
- [35] A. D. Luca, G. Oriolo, and P. R. Giordano. Feature depth observation for image-based visual servoing: Theory and experiments. *The International Journal of Robotics Research*, 27(10):1093–1116, 2008. 1
- [36] J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision (ACCV)*, 2018. 2
- [37] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *CoRR*, abs/1703.09312, 2017. 6
- [38] R. Mahony, P. Corke, and F. Chaumette. Choice of image features for depth-axis control in image based visual servo control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 390–395 vol.1, Sept 2002. 2
- [39] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, April 1999. 2
- [40] K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taix, D. Cremers, and L. V. Gool. Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 2
- [41] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake. Label fusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, May 2018. 2
- [42] G. L. Mariottini, G. Oriolo, and D. Prattichizzo. Image-based visual servoing for nonholonomic mobile robots using epipolar geometry. *IEEE Transactions on Robotics*, 23(1):87–100, Feb 2007. 1
- [43] A. McFadyen, M. Jabeur, and P. Corke. Image-based visual servoing with unknown point feature correspondence. *IEEE Robotics and Automation Letters*, 2(2):601–607, April 2017. 1
- [44] A. Milan, T. Pham, K. Vijay, D. Morrison, A. W. Tow, L. Liu, J. Erskine, R. Grinover, A. Gurman, T. Hunn, N. Kelly-Boxall, D. Lee, M. McTaggart, G. Rallos, A. Razjigaev, T. Rowntree, T. Shen, R. Smith, S. Wade-McCue, Z. Zhuang, C. Lehnert, G. Lin, I. Reid, P. Corke, and J. Leitner. Semantic segmentation from limited training data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1908–1915, May 2018. 3
- [45] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim. Fast video object segmentation by reference-guided mask propagation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [46] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 2
- [47] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 6
- [48] J. A. Piepmeyer, G. V. McMurray, and H. Lipkin. Uncalibrated dynamic visual servoing. *IEEE Transactions on Robotics and Automation*, 20(1):143–147, Feb 2004. 4
- [49] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbelaez, A. Sorkine-Hornung, and L. V. Gool. The 2017 DAVIS challenge on video object segmentation. *CoRR*, abs/1704.00675, 2017. 1
- [50] R. Spica, P. R. Giordano, and F. Chaumette. Coupling active depth estimation and visual servoing via a large projection operator. *The International Journal of Robotics Research*, 36(11):1177–1194, 2017. 2
- [51] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *British Machine Vision Conference (BMVC)*, 2017. 2
- [52] S. Wade-McCue, N. Kelly-Boxall, M. McTaggart, D. Morrison, A. W. Tow, J. Erskine, R. Grinover, A. Gurman, T. Hunn, D. Lee, A. Milan, T. Pham, G. Rallos, A. Razjigaev, T. Rowntree, R. Smith, K. Vijay, Z. Zhuang, C. F. Lehnert, I. D. Reid, P. I. Corke, and J. Leitner. Design of a multi-modal end-effector and grasping system: How integrated design helped win the amazon robotics challenge. *CoRR*, abs/1710.01439, 2017. 3
- [53] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2
- [54] Y. Wang, H. Lang, and C. W. de Silva. A hybrid visual servo controller for robust grasping by wheeled mobile robots. *IEEE/ASME Transactions on Mechatronics*, 15(5):757–769, Oct 2010. 1
- [55] S. Wehrwein and R. Szeliski. Video segmentation with background motion models. In *British Machine Vision Conference (BMVC)*, 2017. 2
- [56] G. Wei, K. Arbter, and G. Hirzinger. Real-time visual servoing for laparoscopic surgery. controlling robot motion with color image segmentation. *IEEE Engineering in Medicine and Biology Magazine*, 16(1):40–45, Jan 1997. 1
- [57] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. S. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *CoRR*, abs/1809.03327, 2018. 1
- [58] U. Yamaguchi, F. Saito, K. Ikeda, and T. Yamamoto. Hsr, human support robot as research and development platform.

The Abstracts of the international conference on advanced mechatronics : toward evolutionary fusion of IT and mechatronics : ICAM, 2015.6:39–40, 2015. 3

- [59] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase. Development of human support robot as the research platform of a domestic mobile manipulator. *ROBOMECH Journal*, 6(1):4, Apr 2019. 3
- [60] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [61] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morona, P. Q. Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 2
- [62] Y. Zuo, W. Qiu, L. Xie, F. Zhong, Y. Wang, and A. L. Yuille. Craves: Controlling robotic arm with a vision-based economic system. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2