2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Perception-Control Coupled Visual Servoing for Textureless Objects Using Keypoint-Based EKF

Allen Tao[1,2*], Jun Yang[1*†], Stanko Oparnica[1], and Wenjie Xue[1]

***Abstract*— Visual servoing is fundamental to robotic applications, enabling precise positioning and control. However, applying it to textureless objects remains a challenge due to the absence of reliable visual features. Moreover, adverse visual conditions, such as occlusions, often corrupt visual feedback, leading to reduced accuracy and instability in visual servoing. In this work, we build upon learning-based keypoint detection for textureless objects and propose a method that enhances robustness by tightly integrating perception and control in a closed loop. Specifically, we employ an Extended Kalman Filter (EKF) that integrates per-frame keypoint measurements to estimate 6D object pose, which drives pose-based visual servoing (PBVS) for control. The resulting camera motion, in turn, enhances the tracking of subsequent keypoints, effectively closing the perception-control loop. Additionally, unlike standard PBVS, we propose a probabilistic control law that computes both camera velocity and its associated uncertainty, enabling uncertainty-aware control for safe and reliable operation. We validate our approach on real-world robotic platforms using quantitative metrics and grasping experiments, demonstrating that our method outperforms traditional visual servoing techniques in both accuracy and practical application.**

I. INTRODUCTION

Visual servoing (VS) refers to the use of visual feedback to control robot motion and serves as a key technique in high-precision tasks. Common approaches, such as image-based visual servo (IBVS), pose-based visual servo (PBVS) and hybrid approaches [1], rely on accurate 2D–2D or 2D–3D keypoint correspondences to estimate the image Jacobian or camera pose for control. Classical VS approaches typically use hand-crafted features [2] to extract these correspondences. However, textureless objects, which are commonly found in industrial environments, pose significant challenges. The absence of distinctive visual features makes establishing reliable correspondences difficult, ultimately degrading servoing performance.

With advances in deep learning, many methods have been developed to reduce classical visual servoing's reliance on hand-crafted features [3], [4], [5], [6], [7], [8]. Early works [3], [4] employ convolutional neural networks (CNNs) to estimate relative camera pose and integrate this into a PBVS framework. However, these approaches are typically trained in specific scenes and often struggle to generalize to unseen environments. More recent approaches use CNNs [6], [7] or Transformer-based networks [9], [8] to improve the reliability of 2D correspondence extraction on textureless objects, and incorporate these features into an IBVS controller. Despite these advances, IBVS suffers from inherent limitations, such as susceptibility to local minima and a limited convergence basin.

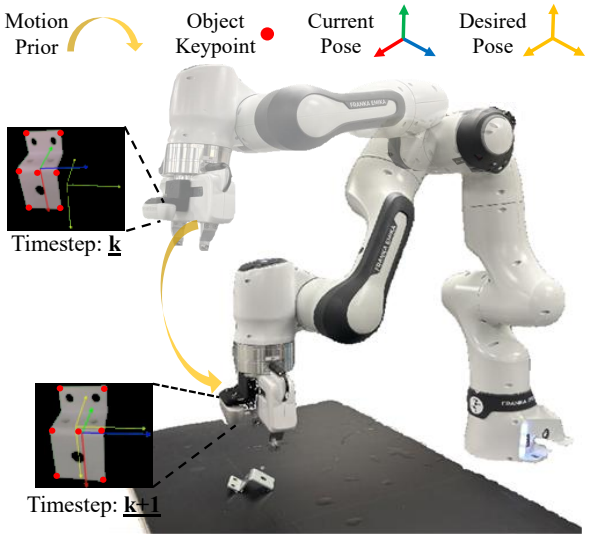


Fig. 1: Perception–control coupled visual servoing framework. An EKF integrates complementary information from keypoints and the motion prior, producing reliable 6D object poses for visual servoing.

On the other hand, recent progress in RGB-based object pose estimation [10], [11], [12], [13], [14], [15], [16] provides a promising alternative for enabling PBVS on textureless objects. These methods assume access to a known 3D object model and estimate the 6D object pose from a single RGB image. Although they can be seamlessly integrated into PBVS frameworks, they typically decouple perception from control and rely solely on single-frame predictions, thereby neglecting temporal information. As a result, they often fail under adverse visual conditions, such as occlusions or abrupt illumination changes, leading to unstable control and reduced robustness in visual servoing. Moreover, most pose estimators [10], [11], [14], [15] produce only a single deterministic pose estimate, which results in fixed control velocities that do not account for uncertainty. To ensure safe and robust robotic operation, it is crucial to model these uncertainties and leverage them to regulate control.

To overcome these limitations, we propose a percep-

*Equal contribution.

†Project lead and corresponding author.

[1]The authors are with Epson Canada Ltd., Toronto, Canada. {Jun.Yang, Stanko.Oparnica, Mark.Xue}@ea.epson.com

[2]The author is with the University of Toronto, Toronto, Canada. This work was done during an internship at Epson Canada Ltd. allen.tao@robotics.utias.utoronto.ca

This work was supported by Epson Canada Ltd.

tion–control coupled framework that leverages temporal information to enable robust visual servoing of textureless objects and maintain performance under adverse conditions. As illustrated in Figure 1, the framework employs an Extended Kalman Filter (EKF) to integrate complementary information from keypoint observations and the motion prior, producing reliable 6D object poses for pose-based visual servoing (PBVS). Unlike many EKF-based trackers that rely on additional sensors such as IMU [17] or incorporate velocity into the state vector [18], [19], our method derives its motion prior directly from the camera velocity computed within the visual servoing loop.

Figure 2 shows the key steps of our framework. We first employ PVNet [13], a state-of-the-art approach, to acquire per-frame object keypoints. Our framework then operates in a closed-loop perception–control cycle: during perception, we use the EKF to integrate keypoints and the motion prior to output a reliable 6D pose; during control, the estimated pose is fed into a probabilistic control law that outputs both the camera velocity and its associated uncertainty; finally, the predicted velocity is used to actuate the robot and propagated into the next perception stage for state prediction. Importantly, the velocity uncertainty computed in the control stage can be leveraged to enhance operational safety, e.g., for collision avoidance.

In summary, our key contributions are:

- We propose a perception-control coupled visual servoing framework designed for textureless objects. By leveraging temporal information, our approach enhances PBVS accuracy while preserving robust performance under challenging conditions, such as occlusions or lighting changes.
- To account for measurement uncertainties, we formulate a probabilistic control law that incorporates this uncertainty into the computed velocity commands, enhancing the safety of robotic operations.

II. RELATED WORK

A. Visual Servoing

Visual servoing (VS) is a control strategy that guides a robot's motion using visual feedback. It is commonly divided into two types based on how the error is defined [1]. In IBVS, the error is defined in image space, while in PBVS, it is defined in Euclidean space. IBVS computes the control velocity either from keypoint correspondences [20], [21], or directly from the pixel intensities of the entire image [22], [23], [24]. PBVS utilizes the relative pose between current and desired camera pose to compute the velocity control command [25]. Generally, PBVS offers a larger convergence region and more efficient 3D trajectories than IBVS, but it is more computationally demanding due to the need for 6D pose estimation at each frame.

Recently, VS has increasingly adopted deep learning techniques to enhance performance. Some studies use deep learning to improve keypoint estimation, thereby facilitating IBVS [6], [7], [8], [9]. For PBVS, methods such as [3], [4] use CNNs to estimate the relative camera pose, which is then integrated into a PBVS framework. Another line of work replaces the velocity control law with neural networks [26], [5], [27], [28], but sacrifices theoretical guarantees like convergence and trajectory optimality in PBVS. In contrast, our approach builds upon the PBVS formulation while incorporating deep learning techniques to improve performance without discarding these guarantees.

B. Object Pose Estimation

Advances in object pose estimation have made visual servoing for textureless objects increasingly feasible. Given a known 3D model, RGB-based pose estimators [10], [12], [13], [11], [14] can recover the object's 6D pose, which can then be directly incorporated into a PBVS framework. Leveraging CNNs or Vision Transformers (ViTs), these approaches either regress the 6D pose directly from a single RGB image [10], [11], [14] or estimate it by first predicting 2D–3D correspondences followed by a PnP algorithm [12], [13].

While single-view pose estimators can be effective, they often fail in the presence of occlusions or lighting variations. To improve robustness, recent works incorporate temporal information for pose estimation [29], [30], [19], [31], [32]. These methods either assume known camera poses [29], [32], [16], or jointly estimate both camera and object poses, a strategy known as object-level SLAM [30], [19], [31]. Temporal information enables more consistent and accurate pose estimates, especially in challenging conditions such as occlusion, motion blur, or weak textures. In this work, we leverage temporal cues through a keypoint-based EKF to estimate robust 6D object poses, enabling reliable visual servoing.

III. FRAMEWORK OVERVIEW

The proposed framework, illustrated in Figure 2, operates in a closed-loop fashion, executing a cycle of perception and control at every frame. At the perception stage (Section IV), we aim to estimate the 6-DOF object pose, $\boldsymbol{T}_{co}$, which represents the transformation from the object frame $\mathcal{F}_o$ to the camera frame $\mathcal{F}_c$:

$$\boldsymbol{T}_{co} = \begin{bmatrix} \mathbf{C}_{co} & \mathbf{t}_{co} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathbb{SE}(3), \tag{1}$$

where $\mathbf{C}_{co} \in \mathbb{SO}(3)$ denotes the object orientation with respect to the camera frame, and $\mathbf{t}_{co} \in \mathbb{R}^3$ represents the object position. This pose estimate $\boldsymbol{T}_{co}$ is then passed to the control stage (Section V), which computes the camera velocity $\mathbf{v}_c$:

$$\mathbf{v}_c = [\mathbf{v}_p \quad \boldsymbol{\omega}] \in \mathbb{R}^6, \tag{2}$$

with $\mathbf{v}_p \in \mathbb{R}^3$ as the translational velocity and $\boldsymbol{\omega} \in \mathbb{R}^3$ as the angular velocity, both expressed in the camera frame $\mathcal{F}_c$. Finally, this estimated velocity is fed back as the motion prior for the next perception stage, completing the closed loop.

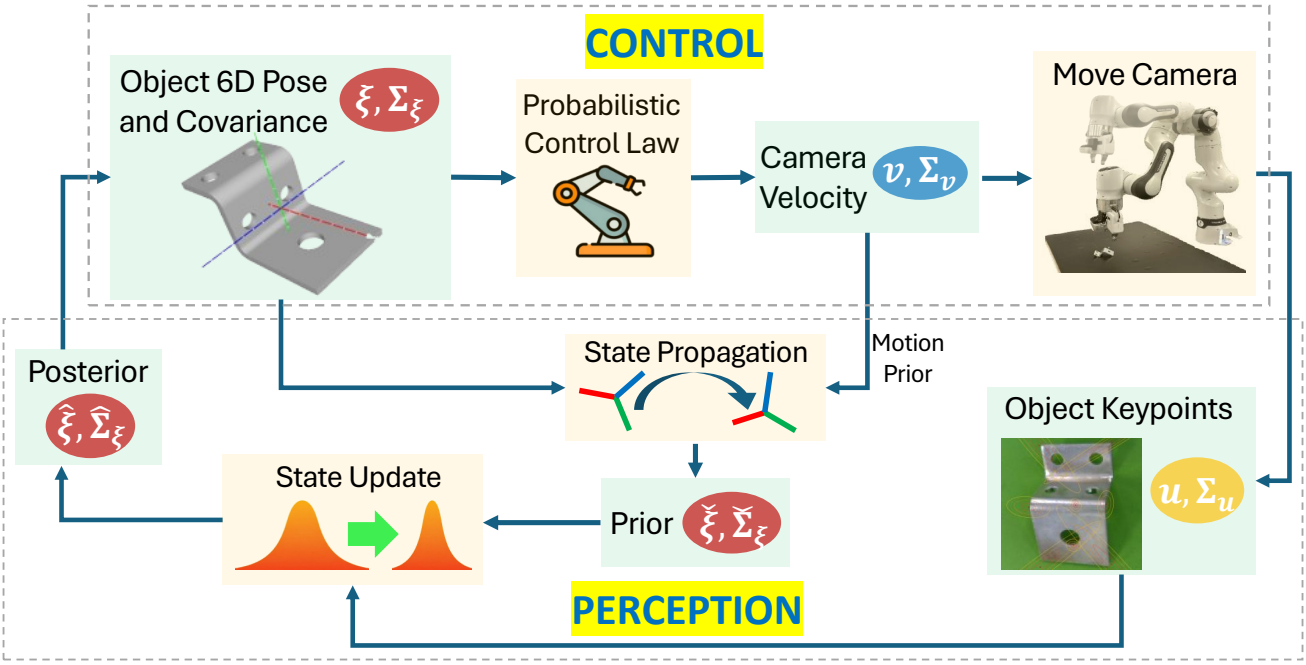


Fig. 2: Our framework operates in a closed-loop cycle: during the perception stage, the EKF fuses keypoints and the motion prior to estimate 6D poses; during the control stage, a probabilistic control law computes camera velocity, which actuates the robot and serves as the motion prior for the next perception stage.

IV. PERCEPTION

The goal of the perception stage is to estimate the 6-DOF object pose, $\boldsymbol{T}_{co}$. We achieve this using a keypoint-based Extended Kalman Filter (EKF), which fuses the motion prior from the previous frame with keypoint measurements from the current frame. The EKF then performs state prediction (Section IV-A) and update (Section IV-B) to produce a refined pose estimate.

A. State Representation and Propagation

State Propagation. We represent the EKF state vector, $\mathbf{x}$, using the 6-DOF object pose:

$$\mathbf{x} = [\mathbf{t}_{co}\ ,\ \mathbf{C}_{co}]\,, \tag{3}$$

where $\mathbf{t}_{co}$ and $\mathbf{C}_{co}$ are defined in Equation 1. For brevity, we will omit the subscript "co" in the following sections. To predict the state at the next timestep, $k{+}1$, we propagate the current state, $\mathbf{x}_k$, using a discrete-time constant-velocity motion model, incorporating the camera's translational velocity $\mathbf{v}_{p,k}$ and angular velocity $\boldsymbol{\omega}_k$:

$$\check{\mathbf{t}}_{k+1} = \exp\left(-\boldsymbol{\omega}_k^{\wedge}\Delta t\right)\mathbf{t}_k - \mathbf{v}_{p,k}\Delta t, \tag{4}$$

$$\check{\mathbf{C}}_{k+1} = \exp\left(-\boldsymbol{\omega}_k^{\wedge}\Delta t\right)\mathbf{C}_k, \tag{5}$$

where $\check{(\cdot)}$ denotes propagated quantities. Δt is the time interval between consecutive time steps, and $\wedge$ represents the skew-symmetric operator applied to a vector. The acquisition of the camera's velocity $\mathbf{v}_{c,k} = [\mathbf{v}_{p,k},\ \boldsymbol{\omega}_k]$ at timestep k will be described in Section V.

Error State Propagation. Alongside the state mean, the EKF must also propagate the associated uncertainty. To achieve this, we define an error state in which the rotational component is linearized using the Lie algebra $\mathfrak{so}(3)$. The resulting 6-dimensional error state vector, $\delta\mathbf{x}$, is defined as:

$$\delta\mathbf{x} = [\delta\mathbf{t}\ ,\ \delta\boldsymbol{\phi}]\,, \tag{6}$$

where $\delta\mathbf{t} \in \mathbb{R}^3$ and $\delta\boldsymbol{\phi} \in \mathbb{R}^3$ represent the translational and rotational errors, respectively.

By linearizing the motion model in Equations (4) and (5) around the current state estimate, the discrete-time error-state propagation is given by:

$$\begin{bmatrix}\delta\check{\mathbf{t}}_{k+1}\\ \delta\check{\boldsymbol{\phi}}_{k+1}\end{bmatrix} = \mathbf{F}\begin{bmatrix}\delta\mathbf{t}_k\\ \delta\boldsymbol{\phi}_k\end{bmatrix} + \mathbf{G}\mathbf{n}, \tag{7}$$

where $\mathbf{n} = \begin{bmatrix}\mathbf{n_t}^T & \mathbf{n}_{\boldsymbol{\phi}}{}^T\end{bmatrix}^T$ represents the camera velocity noise, and $\mathbf{G}$ is the noise Jacobian. The error-state transition matrix, $\mathbf{F}_k$, is:

$$\mathbf{F}_k = \begin{bmatrix}\frac{\partial \exp\left(-\boldsymbol{\omega}_k^{\wedge}\Delta t\right)\mathbf{t}_k}{\partial \mathbf{t}_k} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \frac{\partial \ln\left(\exp\left(-\boldsymbol{\omega}_k^{\wedge}\Delta t\right)\mathbf{C}_k\right)^{\vee}}{\partial \boldsymbol{\phi}_k}\end{bmatrix}, \tag{8}$$

with:

$$\frac{\partial \exp\left(-\boldsymbol{\omega}_k^{\wedge}\Delta t\right)\mathbf{t}_k}{\partial \mathbf{t}_k} = \exp\left(-\boldsymbol{\omega}_k^{\wedge}\Delta t\right), \tag{9}$$

$$\frac{\partial \ln(\exp\left(-\boldsymbol{\omega}_k^{\wedge}\Delta t\right)\mathbf{C}_k)^{\vee}}{\partial \boldsymbol{\phi}_k} = \mathcal{J}_r^{-1}\left(\ln(\exp\left(-\boldsymbol{\omega}_k^{\wedge}\Delta t\right)\mathbf{C}_k)^{\vee}\right). \tag{10}$$

Here, $\ln(\cdot)^{\vee}$ denotes the logarithmic mapping from Lie group $\mathbb{SO}(3)$ to its Lie algebra $\mathfrak{so}(3)$. The operator $\mathcal{J}_r(\cdot)$ is the right Jacobian. This Jacobian is derived using the

right perturbation model together with the Baker-Campbell-Hausdorff (BCH) approximation. For detailed derivations, readers are referred to [33].

Finally, the state covariance $\mathbf{P}$ is propagated to the next timestep, $k+1$, as follows:

$$\mathbf{R} = \text{diag}\left(\sigma_{v_p}\mathbf{I}\,,\ \sigma_{v_\omega}\mathbf{I}\right), \tag{11}$$

$$\check{\mathbf{P}}_{k+1} = \mathbf{F}_k\mathbf{P}_k\mathbf{F}_k^T + \mathbf{G}_k\mathbf{R}\mathbf{G}_k^T\Delta t, \tag{12}$$

where σ_{v_p} and σ_{v_ω} denote the standard deviations of the translational and angular components of the camera velocity noise, respectively, and $\mathbf{G}_k$ is the identity noise Jacobian.

B. Measurement Model and Update

While the propagation step provides a prior estimate of the pose, it is susceptible to drift over time. To correct this, the proposed EKF performs measurement updates using 2D–3D keypoint correspondences obtained from per-frame RGB images.

1) ***Keypoint Measurements****:* In the update step, the measurement comprises 2D image keypoints associated with 3D object points $\mathbf{X}_o$. For each object, we first select N 3D keypoints on the CAD model using farthest point sampling (FPS). At runtime, the corresponding 2D keypoints are estimated using PVNet [13], which predicts their image locations and provides an associated covariance matrix for each keypoint to model localization uncertainty. For timestep k+1, the measured keypoints and their uncertainties are stacked into the measurement vector $\mathbf{z}_{k+1}$ and covariance matrix $\mathbf{Q}_{k+1}$, defined as:

$$\mathbf{z}_{k+1} = \begin{bmatrix}\mathbf{u}_0\\ \vdots \\ \mathbf{u}_{N-1}\end{bmatrix}\ ,\ \mathbf{Q}_{k+1} = \begin{bmatrix}\boldsymbol{\Sigma}_{\mathbf{u}_0} & & \\ & \ddots & \\ & & \boldsymbol{\Sigma}_{\mathbf{u}_{N-1}}\end{bmatrix}, \tag{13}$$

where $\mathbf{u}_i \in \mathbb{R}^2$ denotes the *measured* 2D keypoint of the i-th object point and $\boldsymbol{\Sigma}_{\mathbf{u}_i} \in \mathbb{R}^{2\times 2}$ its covariance. Figure 3 shows an example of the 3D object points $\mathbf{X}_o$, along with the measured keypoints, $\mathbf{u}_i$, and their associated covariance matrices, $\boldsymbol{\Sigma}_{\mathbf{u}_i}$. Note that while we adopt PVNet, other keypoint detection methods [12], [31] can also be integrated into our EKF framework.

2) ***Measurement Model****:* We define the measurement model $h(\cdot)$ by first transforming each 3D object point $\mathbf{X}_{o,i} \in \mathbb{R}^3$ from the object frame $\mathcal{F}_o$ to the camera frame $\mathcal{F}_c$ using the object pose estimate $\left(\check{\mathbf{C}}_{k+1}, \check{\mathbf{t}}_{k+1}\right)$, and then projecting the result onto the image plane:

$$\check{\mathbf{u}}_i = h\left(\check{\mathbf{C}}_{k+1}, \check{\mathbf{t}}_{k+1}\right) \tag{14}$$

$$= \text{proj}\left(\check{\mathbf{C}}_{k+1}\mathbf{X}_{o,i} + \check{\mathbf{t}}_{k+1}\right) \tag{15}$$

where $\check{\mathbf{u}}_i \in \mathbb{R}^2$ is the *predicted* 2D location of the i-th object point, and $\text{proj}(\cdot)$ denotes the perspective projection function. The residual error $\boldsymbol{\epsilon}_{k+1}$ at frame $k+1$ is defined as the difference between the measured and predicted keypoints:

$$\boldsymbol{\epsilon}_{k+1} = \mathbf{z}_{k+1} - \check{\mathbf{z}}_{k+1} = \begin{bmatrix}\mathbf{u}_0 - \check{\mathbf{u}}_0\\ \vdots \\ \mathbf{u}_{N-1} - \check{\mathbf{u}}_{N-1}\end{bmatrix}, \tag{16}$$

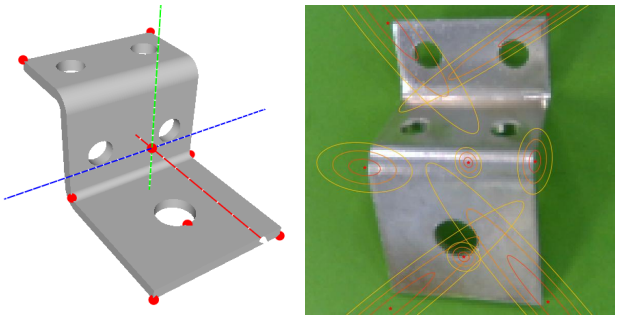


Fig. 3: Example of 3D object points $\mathbf{X}_o$, along with 2D image keypoints, $\mathbf{u}_i$, and their associated uncertainties, $\boldsymbol{\Sigma}_{\mathbf{u}_i}$, as estimated using PVNet [13].

where $\mathbf{z}_{k+1} = [\mathbf{u}_0^\top, \ldots, \mathbf{u}_{N-1}^\top]^\top$ is the vector of measured keypoints from PVNet, and $\check{\mathbf{z}} = [\check{\mathbf{u}}_0^\top, \ldots, \check{\mathbf{u}}_{N-1}^\top]^\top$ is the corresponding vector of predicted keypoints.

To acquire the measurement Jacobian $\mathbf{H}_i$ for each keypoint residual $\boldsymbol{\epsilon}_i$, we differentiate the residual $\boldsymbol{\epsilon}_i$ with respect to the error state $\delta\mathbf{x}$:

$$\mathbf{H}_i = \frac{\partial \boldsymbol{\epsilon}_i}{\partial \delta\mathbf{x}} = \begin{bmatrix}\frac{\partial \boldsymbol{\epsilon}_i}{\partial \delta\mathbf{t}_{k+1}} & \frac{\partial \boldsymbol{\epsilon}_i}{\partial \delta\boldsymbol{\phi}_{k+1}}\end{bmatrix}, \tag{17}$$

where

$$\frac{\partial \boldsymbol{\epsilon}_i}{\partial \delta\mathbf{t}_{k+1}} = \frac{\partial\left(\mathbf{u}_i - \check{\mathbf{u}}_i\right)}{\partial \delta\mathbf{t}_{k+1}} = -\frac{\partial h\left(\mathbf{C}_{k+1}, \mathbf{t}_{k+1}\right)}{\partial \delta\mathbf{t}_{k+1}}, \tag{18}$$

$$\frac{\partial \boldsymbol{\epsilon}_i}{\partial \delta\boldsymbol{\phi}_{k+1}} = \frac{\partial\left(\mathbf{u}_i - \check{\mathbf{u}}_i\right)}{\partial \delta\boldsymbol{\phi}_{k+1}} = -\frac{\partial h\left(\mathbf{C}_{k+1}, \mathbf{t}_{k+1}\right)}{\partial \delta\boldsymbol{\phi}_{k+1}}. \tag{19}$$

To derive $\frac{\partial h(\mathbf{C}_{k+1}, \mathbf{t}_{k+1})}{\partial \delta\boldsymbol{\phi}_{k+1}}$ in Equation 19, we employ a left perturbation model in the Lie algebra $\mathfrak{so}(3)$, which represents the rotational update as a local increment on the manifold and enables its linearization for differentiation. The overall measurement Jacobian $\mathbf{H}_{k+1}$ at frame k+1 is finally formed by stacking the individual Jacobians $\mathbf{H}_{i,k+1}$ for all keypoints:

$$\mathbf{H}_{k+1} = [\mathbf{H}_0, \mathbf{H}_1, \cdots, \mathbf{H}_{N-1}]^T \tag{20}$$

3) ***EKF Update****:* Once we obtained the measurement Jacobian $\mathbf{H}_{k+1}$, we can update our estimate using a regular EKF update:

$$\begin{aligned}\mathbf{K}_{k+1} &= \check{\mathbf{P}}_{k+1}\mathbf{H}_{k+1}^T\left(\mathbf{H}_{k+1}\check{\mathbf{P}}_{k+1}\mathbf{H}_{k+1}^T + \mathbf{Q}_{k+1}\right)^{-1}\\ \hat{\mathbf{P}}_{k+1} &= \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\right)\check{\mathbf{P}}_{k+1}\\ \delta\mathbf{x}_{k+1} &= \mathbf{K}_{k+1}\boldsymbol{\epsilon}_{k+1}\end{aligned} \tag{21}$$

where $\delta\mathbf{x}_{k+1} = [\delta\mathbf{t}_{k+1}\ ,\ \delta\boldsymbol{\phi}_{k+1}]$ is state update, to inject $\delta\mathbf{x}_{k+1}$ into the predicted nominal states, we apply:

$$\begin{aligned}\hat{\mathbf{t}}_{k+1} &= \check{\mathbf{t}}_{k+1} + \delta\mathbf{t}_{k+1}\\ \hat{\mathbf{C}}_{k+1} &= \exp\left(\delta\boldsymbol{\phi}_{k+1}^\wedge\right)\check{\mathbf{C}}_{k+1}\end{aligned} \tag{22}$$

where $\exp(\cdot)$ is the exponential mapping from the Lie algebra $\mathfrak{so}(3)$ to its Lie group $\mathbb{SO}(3)$, and $\hat{(\cdot)}$ denotes the updated quantities.

V. PROBABILISTIC CONTROL

For each frame k, given the updated EKF state $\hat{\mathbf{x}}$ (from Equation 22), the objective of the control stage is to compute the camera velocity $\mathbf{v}_c$ from it. This velocity both actuates the robot and serves as the motion prior for state propagation in the next timestep (Equations 4 and 5), thereby closing the perception–control loop. Additionally, to account for uncertainty in the estimated velocity, we estimate its covariance $\mathbf{\Sigma}_{\mathbf{v}_c}$, enabling uncertainty-aware motion execution.

To compute $\mathbf{v}_c$, we follow the classic PBVS control law, which derives the camera velocity through the desired and current object poses. Specifically, let $\mathbf{T}_{c^*c}$ denote the relative transformation from the current camera frame $\mathcal{F}_c$ to the desired camera frame $\mathcal{F}_{c^*}$, obtained as

$$\mathbf{T}_{c^*c} = \mathbf{T}_{c^*o} \cdot \mathbf{T}_{co}^{-1} = \begin{bmatrix} \mathbf{C}_{c^*c} & \mathbf{t}_{c^*c} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{23}$$

where $\mathbf{T}_{c^*o}$ represents the desired object pose, the current object pose $\mathbf{T}_{co}$ is assembled from the updated EKF states in Equation 22:

$$\mathbf{T}_{co} = \begin{bmatrix} \hat{\mathbf{C}}_{co} & \hat{\mathbf{t}}_{co} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{24}$$

then the mean of camera velocity $\mathbf{v}_c$ is computed using the PBVS control law [1]:

$$\mathbf{v}_c = -\lambda \begin{bmatrix} \mathbf{C}_{c^*c}^T & \mathbf{t}_{c^*c} \\ & \theta\mathbf{u} \end{bmatrix} \tag{25}$$

where $\theta\mathbf{u}$ is the axis–angle representation of the rotation matrix $\mathbf{C}_{c^*c}$, and $\lambda > 0$ is a control gain.

To acquire the velocity uncertainty, $\mathbf{\Sigma}_{\mathbf{v}_c}$, we linearize the system (Equations 23-25) and propagate the EKF state covariance $\hat{\mathbf{P}}$ through the PBVS control law. Specifically, we derive the Jacobian of the camera velocity with respect to the EKF error state vector, denoted as $\mathbf{J}_{\delta\hat{\mathbf{x}}}$:

$$\mathbf{J}_{\delta\hat{\mathbf{x}}} = \frac{\partial \mathbf{v}_c}{\partial \delta\hat{\mathbf{x}}} = \frac{\partial \mathbf{v}_c}{\partial \boldsymbol{\xi}_{c*c}} \frac{\partial \boldsymbol{\xi}_{c*c}}{\partial \delta\hat{\mathbf{x}}} \tag{26}$$

where $\boldsymbol{\xi}_{c*c} \in \mathbb{R}^6$ is the Lie algebra representation of the relative pose $\mathbf{T}_{c^*c}$. The camera velocity covariance is then obtained via forward propagation:

$$\mathbf{\Sigma}_{\mathbf{v}_c} = \mathbf{J}_{\delta\hat{\mathbf{x}}} \hat{\mathbf{P}} \mathbf{J}_{\delta\hat{\mathbf{x}}}^T \tag{27}$$

where $\mathbf{\Sigma}_{\mathbf{v}_c}$ is a 6×6 matrix. To quantify this uncertainty, we express it as the differential entropy:

$$h_e\left(\mathbf{\Sigma}_{\mathbf{v}_c}\right) = \frac{1}{2} \ln\left((2\pi e)^n \left|\mathbf{\Sigma}_{\mathbf{v}_c}\right|\right) \tag{28}$$

where $h_e\left(\mathbf{\Sigma}_{\mathbf{v}_c}\right)$ is expressed in nats. To incorporate the estimated velocity uncertainty into the robot control policy, we define an uncertainty threshold. When the entropy exceeds this threshold, the velocity is significantly reduced to ensure safety. This mechanism is crucial for maintaining safe robotic operation under uncertain motion estimates.

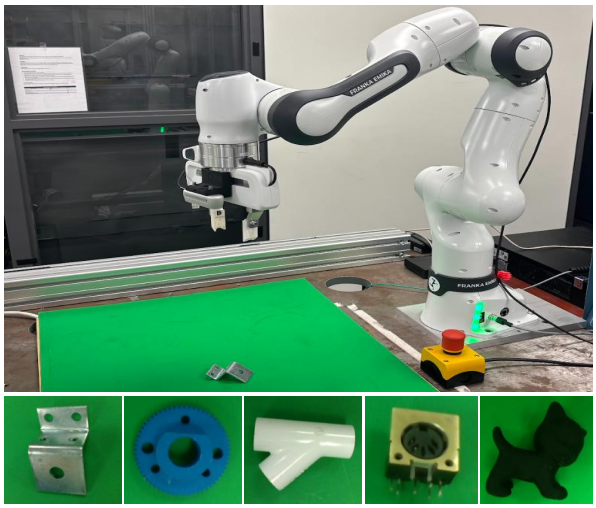


Fig. 4: Experimental setup. Top: Hardware platform for visual servoing. Bottom: Target objects for evaluation.

VI. EXPERIMENTS

A. Experiment Setup

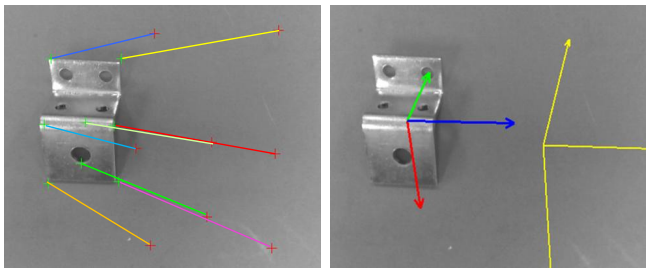
We conduct experiments on a real-world robotic platform, shown in Figure 4. The setup includes a 7-DoF Franka Emika arm with an Intel RealSense D435 camera mounted on the end-effector. A desktop with a 3.60 GHz CPU and NVIDIA A6000 GPU acts as the server. The Franka controller is built on the ViSP library [34], communicating to the server via ZeroMQ [35]. At each timestep, the server processes the camera image and sends velocity commands to the robot for real-time control.

We perform visual servoing experiments on five textureless objects with varied colors and materials (Figure 4). In each episode, the desired camera pose $\mathbf{T}_{c^*o}$ is set roughly 15 cm above the target, with ± 5 cm translational variation in XYZ. The initial pose $\mathbf{T}_{co}$ is sampled around 30 cm above the object, with random translations of $[-10\ cm, 10\ cm]$ and rotations of $[-75°, 75°]$ applied. For each object, we run 30 visual servoing trials using our method and baseline approaches. As shown in Figure 5c, trials are performed under diverse backgrounds, lighting conditions, and partial occlusions. To reflect difficulty, we group scenes into **Normal Conditions** (controlled settings) and **Adverse Conditions** (with challenging backgrounds, lighting, and occlusions).

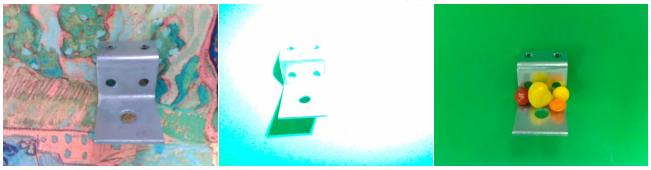
B. Implementation and Baselines

As discussed in Section IV-B.1, we use PVNet [13] to extract 2D keypoints for textureless objects. For each object, PVNet is trained from scratch on approximately 200,000 synthetic images. We quantitatively evaluate our approach against classic IBVS and PBVS baselines. To ensure a fair comparison, we use the same trained PVNet model and integrate it into both baseline methods, referred to as IBVS-PVNet and PBVS-PVNet, respectively.

- **IBVS+PVNet**. Figure 5a shows our integration of PVNet into the IBVS framework. We use the target camera pose $\mathbf{T}_{c^*o}$ to project 3D model points to obtain



(a) IBVS+PVNet (b) PBVS+PVNet



(c) Experiments with different visual backgrounds, lighting conditions, and partial occlusions.

Fig. 5: Baseline approaches used in our evaluation. (a) IBVS+PVNet. (b) PBVS+PVNet. (c) Experiments with different backgrounds, lighting conditions, and occlusions.

desired 2D keypoints $\mathbf{p}^*$. At each timestep, PVNet extracts current 2D keypoints $\mathbf{p}$ from the image. The correspondences $\mathbf{p}, \mathbf{p}^*$ feed into the IBVS controller to compute the velocity command for visual servoing.

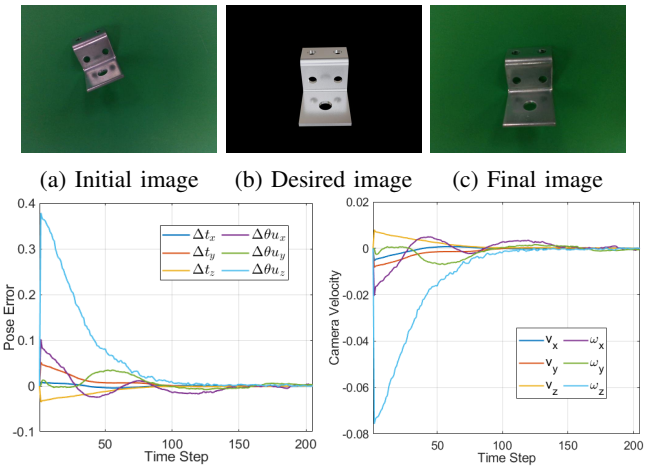
- **PBVS+PVNet**. To implement PBVS, we estimate the current camera pose $\mathbf{T}_{co}$ per frame using the uncertainty-driven PnP algorithm [13] with 2D keypoints from PVNet. The estimated and target poses $\mathbf{T}_{c^*o}$ are used in the PBVS control law [1] to compute the camera velocity. Figure 5b illustrates this process.

C. Evaluation Metrics

Inspired by [26], [36], we use four criteria to quantitatively analyze the servo performance: servo success rate (**SR**), final translation error (**TE**), final rotation error (**RE**), and trajectory length ratio (**LR**).

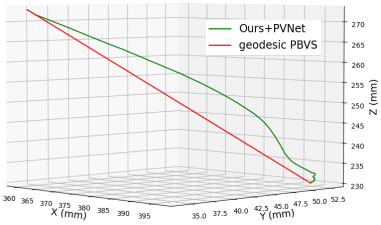
- **Servo Success Rate (SR)**: A trial is successful if final velocities are near zero and the final pose is accurate. We use the average model distance (ADD) metric, considering a pose accurate if ADD is below 10% of the object diameter.
- **Final Translation Error (TE)**: Defined as the norm of the translation component of the relative transformation $\mathbf{T}_{c^*c}$ between final and desired poses.
- **Final Rotation Error (RE)**: The angular difference from the rotation component of $\mathbf{T}_{c^*c}$, computed via axis-angle representation.
- **Trajectory Length Ratio (LR)**: Defined as the fraction of the method's trajectory length divided by the length of the geodesic PBVS trajectory [36]. As illustrated in Figure 6f, The geodesic of PBVS is the shortest path in SE(3) connecting the initial and desired camera poses, representing the most efficient trajectory.

To ensure fair evaluation, we compute the averages of translation error (**TE**), rotation error (**RE**), and trajectory length ratio (**LR**) only over successful servoing trajectories.



(a) Initial image (b) Desired image (c) Final image

(d) Pose error (e) Camera velocity



(f) 3D trajectories

Fig. 6: Example visual servoing experiment on the object "Zigzag": (a) Initial image. (b) Desired image (rendered with desired pose). (c) Final image after servoing. (d) Pose error over time (units: meters and radians). (e) Camera velocity commands (units: meters and radians). (f) 3D trajectories of our approach and geodesic PBVS.

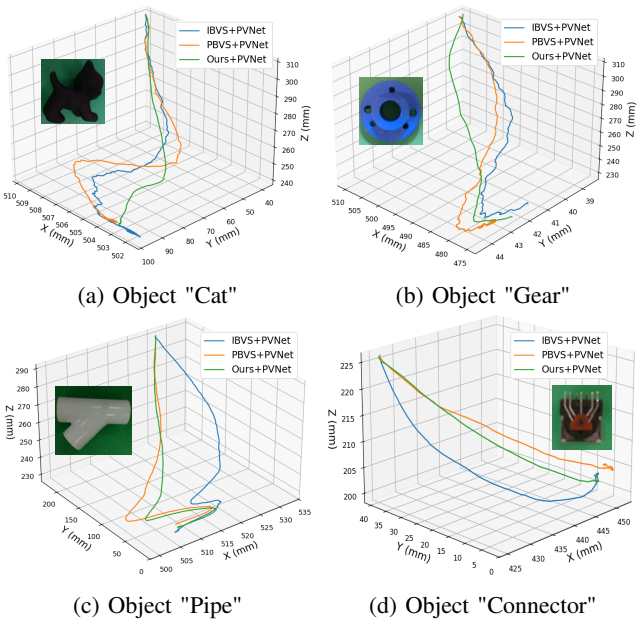
D. Experiment Results

We first present an example of our perception-control coupled visual servoing in Figure 6. The initial and desired image, depicted in Figures 6a and 6b, reveal a significant initial pose discrepancy. As the servoing progresses, our method rapidly reduces the pose error and ensures a smooth decrease in camera velocity. Ultimately, the pose error is minimized, and the camera velocity converges to zero, demonstrating the effectiveness of our method. This behavior results in a stable and smooth camera trajectory (Figure 6f) that closely resembles the geodesic path of a PBVS strategy.

Table I presents the servoing performance of different approaches evaluated with multiple trials. Our method consistently outperforms both IBVS+PVNet and PBVS+PVNet in success rate (**SR**), achieving **95.12%** in normal conditions and **82.61%** in adverse conditions, demonstrating superior robustness. Moreover, our approach produces the lowest translation error (**TE**), rotation error (**RE**), and length ratio (**LR**) in both scenarios, reducing these metrics by up to 0.92 mm, 0.45 degrees, and 0.5, respectively. These findings highlight the advantages of our method in enhancing accuracy, efficiency, and reliability for visual servoing with textureless objects in real-world environments.

Metric	Normal Conditions			Adverse Conditions		
	IBVS+PVNet	PBVS+PVNet	Ours+PVNet	IBVS+PVNet	PBVS+PVNet	Ours+PVNet
SR (%)	87.81	84.15	**95.12**	52.17	40.58	**82.61**
TE (mm)	3.27 ± 1.72	3.66 ± 1.76	**3.17 ± 1.45**	4.91 ± 4.09	5.53 ± 4.08	**3.99 ± 2.72**
RE (deg)	3.91 ± 2.69	3.98 ± 3.81	**3.81 ± 2.68**	6.15 ± 3.98	6.44 ± 4.16	**5.70 ± 3.98**
LR (\)	1.25 ± 0.36	1.29 ± 0.58	**1.11 ± 0.15**	1.68 ± 0.78	1.92 ± 1.02	**1.18 ± 0.23**

TABLE I: Real-world performance comparison of visual servoing approaches. Bold values indicate the best performance. Errors are reported as mean ± standard deviation over successful trials. Note that, for our approach, the results were obtained without activating the uncertainty-aware velocity reduction, allowing a fair comparison of standard servoing performance.



(a) Object "Cat" (b) Object "Gear"

(c) Object "Pipe" (d) Object "Connector"

Fig. 7: Comparison of 3D trajectories during visual servoing on different objects using IBVS, PBVS, and our method.

Figure 7 further compares the 3D trajectories of our approach against baseline methods, using the same PVNet measurements. Compared to IBVS and PBVS, the trajectories generated by our method are shorter and smoother, further demonstrating its effectiveness and stability.

E. Uncertainty Evaluation

To evaluate the estimated velocity uncertainties, we compared the error in our estimated velocities against their corresponding uncertainty estimates. For each frame, we manually labeled the object pose, $\mathbf{T}_{co}^{GT}$, and computed the ground truth camera velocity, $\mathbf{v}_c^{GT}$, using the labeled pose, $\mathbf{T}_{co}^{GT}$, and the desired camera pose, $\mathbf{T}_{c^*o}$. As illustrated in Figure 8, the estimated uncertainties closely follow the actual velocity errors, indicating that the uncertainty estimates are both reliable and well aligned with our goal.

F. Integration in a Grasping Pipeline

We further evaluate the applicability of our approach in robotic grasping tasks with textureless objects. In this setup, visual servoing first drives the robot to a target object pose.

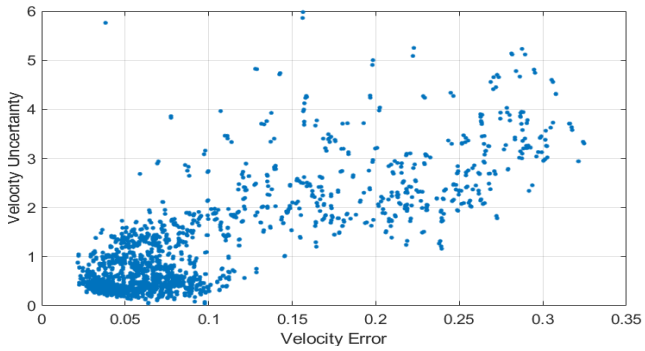


Fig. 8: The predicted velocity uncertainty correlates well with the velocity error. We compute the Pearson correlation coefficient (0.81 in this example).

From this pose, a preprogrammed motion offset, defined as a fixed object-specific rigid transformation, moves the end-effector from the servoing target pose to the grasp pose. The grasp pose is manually specified for each object based on its geometry, and the same object-specific offset is used for all baselines to ensure a fair evaluation.

For each object, we conduct 19–23 servoing trials using both our method and baseline approaches. The lighting conditions, as well as the desired and initial camera poses, match those in Section VI-A, with the initial poses sampled from a broader range of rotations up to $[-105°, 105°]$.

Table II summarizes the grasping results. Our approach consistently outperforms both IBVS and PBVS for all objects, achieving the highest average success rate of 89.9%. These results demonstrate the effectiveness and robustness of our method in real-world robotic grasping scenarios.

VII. CONCLUSION

In this work, we proposed a perception–control coupled visual servoing framework that operates solely on RGB

Method	Zigzag	Pipe	Gear	Cat	Connector	Avg.
IBVS	79.2	69.6	78.3	71.4	73.7	74.4
PBVS	75.0	56.5	86.9	76.2	73.7	73.7
Ours	**95.8**	**78.3**	**95.6**	**90.5**	**89.5**	**89.9**

TABLE II: Robotic grasping success rate (%) using visual servoing across different objects.

images, enabling robust servoing of textureless objects while maintaining performance under adverse conditions. By integrating per-frame object keypoints with motion priors through an Extended Kalman Filter (EKF), the framework achieves accurate and stable 6D pose estimation for effective pose-based visual servoing. We further introduced a probabilistic control law that models velocity command uncertainties, enhancing the safety of robotic operations. Experimental results on real-world robotic systems show that the proposed framework improves the performance of classical visual servoing algorithms, achieving higher accuracy and robustness under diverse conditions.

Future directions include extending the framework to dynamic environments, incorporating active viewpoint selection, and generalizing to more challenging object types such as CAD-less or deformable objects. For example, high uncertainty in the estimated velocity could trigger active perception, enabling motion replanning to better handle occlusions or dynamic obstacles.

ACKNOWLEDGEMENTS

This work was developed with the assistance of OpenAI's GPT-4, which was used to (i) generate and refine code for robotic system integration, and (ii) editing and grammar enhancement for the manuscript. All AI-generated content was carefully reviewed and validated by the authors.

REFERENCES

[1] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.

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

[4] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Training deep neural networks for visual servoing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[5] C. Yu, Z. Cai, H. Pham, and Q.-C. Pham, "Siamese convolutional neural network for sub-millimeter-accurate camera pose estimation and visual servoing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[6] N. Adrian, V.-T. Do, and Q.-C. Pham, "Dfbvs: Deep feature-based visual servo," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022.

[7] J. Luo, L. Zhu, Z. Zhang, and W. Bai, "Uncalibrated 6-dof robotic grasping with rgb-d sensor: A keypoint-driven servoing method," *IEEE Sensors Journal*, 2024.

[8] A. Chen, H. Yu, Y. Wang, and R. Xiong, "Cns: Correspondence encoded neural image servo policy," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[9] A. Scherl, S. Thalhammer, B. Neuberger, W. Wöber, and J. García-Rodríguez, "Vit-vs: On the applicability of pretrained vision transformer features for generalizable visual servoing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[10] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

[11] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *European Conference on Computer Vision (ECCV)*, 2018.

[12] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[13] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern recognition (CVPR)*, 2019.

[14] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic, "Megapose: 6d pose estimation of novel objects via render & compare," in *Conference on Robot Learning (CoRL)*, 2022.

[15] L. Xu, H. Qu, Y. Cai, and J. Liu, "6d-diff: A keypoint diffusion framework for 6d object pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern recognition (CVPR)*, 2024.

[16] J. Yang, W. Xue, S. Ghavidel, and S. L. Waslander, "Active 6d pose estimation for textureless objects using multi-view rgb frames," *The International Journal of Robotics Research*, 2025.

[17] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[18] M. A. Mehralian and M. Soryani, "Ekfpnp: extended kalman filter for camera pose estimation in a sequence of images," *IET Image Processing*, vol. 14, no. 15, pp. 3774–3780, 2020.

[19] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Keypoint-based category-level object pose tracking from an rgb sequence with uncertainty estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[20] J. T. Feddema and O. R. Mitchell, "Vision-guided servoing with feature-based trajectory generation (for robots)," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 5, pp. 691–700, 2002.

[21] T. T. H. Tran and E. Marchand, "Real-time keypoints matching: application to visual servoing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[22] C. Collewet and E. Marchand, "Photometric visual servoing," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 828–834, 2011.

[23] E. Marchand, "Subspace-based direct visual servoing," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2699–2706, 2019.

[24] S. Felton, P. Brault, E. Fromont, and E. Marchand, "Visual servoing in autoencoder latent space," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3234–3241, 2022.

[25] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: keeping the object in the field of vision," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[26] H. Yu, A. Chen, K. Xu, Z. Zhou, W. Jing, Y. Wang, and R. Xiong, "A hyper-network based end-to-end visual servoing with arbitrary desired poses," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, 2023.

[27] E. Y. Puang, K. P. Tee, and W. Jing, "Kovis: Keypoint-based visual servoing with zero-shot sim-to-real transfer for robotics manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[28] S. Felton, E. Fromont, and E. Marchand, "Siame-se (3): regression in se (3) for end-to-end visual servoing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[29] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *European Conference on Computer Vision (ECCV)*, 2020.

[30] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1328–1342, 2021.

[31] N. Merrill, Y. Guo, X. Zuo, X. Huang, S. Leutenegger, X. Peng, L. Ren, and G. Huang, "Symmetry and uncertainty-aware object slam for 6dof object pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern recognition (CVPR)*, 2022.

[32] J. Yang, W. Xue, S. Ghavidel, and S. L. Waslander, "6d pose estimation for textureless objects on rgb frames using multi-view optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[33] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2024.

[34] É. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, 2005.

[35] P. Hintjens, *ZeroMQ: messaging for many applications*. " O'Reilly Media, Inc.", 2013.

[36] S. Felton, E. Fromont, and E. Marchand, "Deep metric learning for visual servoing: when pose and image meet in latent space," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.