2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

From One Image to Precision Pose: Seed-Diverse Diffusion Models and Model-Selection-Driven Hybrid Servoing in Limited Viewpoints

Daigo Terazono[1], Takashi Nammoto[2], Ryota Kato[2], Naoya Chiba[1], Shingo Kagami[1], and Koichi Hashimoto[1]

***Abstract*—In robotic visual servoing, when prior measurement or imaging is impractical, control must rely on a single image of the initial view. Diffusion models can generate 3D shapes from a single image; however, the blind spot area involves uncertainty, which may degrade alignment accuracy if used directly in model-dependent control manipulation.**

This study proposes a hybrid visual servo control method that operates under the assumption of this uncertainty. From a single image, multiple 3D shape candidates are sampled using a pre-trained generative model, and then progressively controlled while retaining them. First, rough positioning is performed using PBVS (Position-Based Visual Servoing) with multiple shape candidates. Next, the system compares the candidate images rendered with the observed image and selects the best model based on geometric error and visual similarity. Finally, IBVS (Image-Based Visual Servoing) uses the selected model to refine slight alignment errors with high precision.

This proposed method achieves high-precision approach and alignment from minimal input of a single image, providing a framework that resolves the problems of shape uncertainty and control error caused by 3D generation. Experiments show that the convergence success rate improved as the number of shape candidates increased and that high-precision alignment was achieved through the staged integration of PBVS and IBVS.

I. Introduction

Grasping and manipulating unknown objects requires understanding their 3D shape and dimensions. When prior measurement or multi-view imaging is difficult, the 3D model or reference images of the target object cannot be obtained beforehand, resulting in insufficient information for visual guidance [1]. Such situations are common in home robot control, food picking, and harsh environment tasks, forcing robots to rely on limited initial images. Therefore, methods are needed to estimate the 3D shape and pose of the target object from limited images and incorporate this into robot control.

A. Generating 3D Models from Few Images Using Diffusion Models

As a method to cover the lack of prior information, techniques for generating 3D models of objects from a single image have gained attention in recent years, and approaches using diffusion models have been particularly reported [2,3]. Even in situations where prior information cannot be prepared, obtaining high-quality 3D models in a single shot can supplement the visual information of a robot.

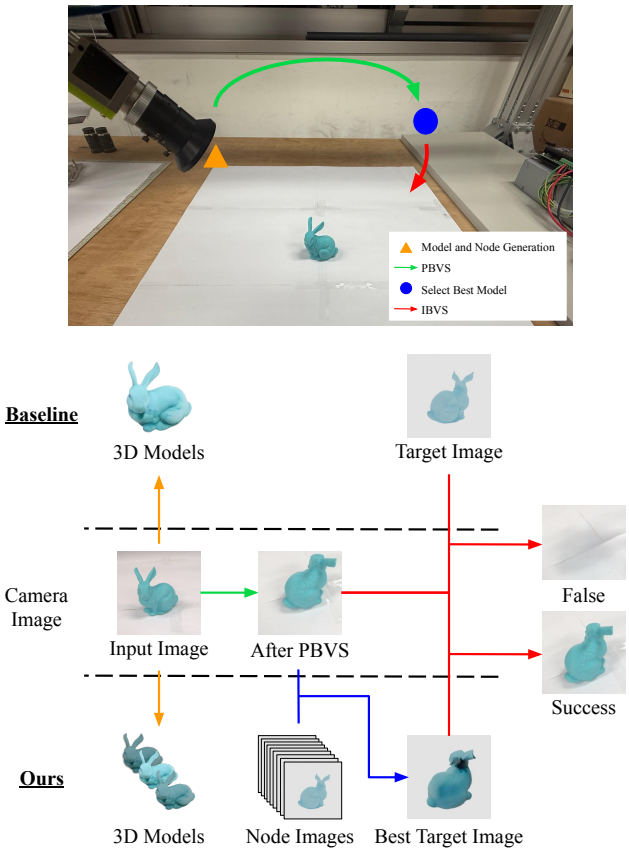


Fig. 1: Overview of the proposed framework: A model selection mechanism that mitigates shape uncertainty in 3D generative models and a hybrid visual servoing system for high-precision pose alignment.

However, observations obtained from a single viewpoint have many blind spots. 3D models generated from a single point of view involve shape uncertainty because they supplement the blind spots based on training data. Indeed, Jiang et al. report that under single-image conditions, approximately 45% of point clouds become unobserved regions, compromising consistency [3]. Furthermore, since the shapes generated by generative models vary with each sampling, there is a risk that shape errors on the object's backside in particular may cause model-dependent control accuracy

[1]Authors are with the Department of System Information Sciences, Graduate School of Information Sciences, Tohoku University, Aoba-ku, Sendai 980-8579, Japan (terazono.daigo.p4@dc.tohoku.ac.jp; chiba@nchiba.net; swk@ic.is.tohoku.ac.jp; koichi.hashimoto.a8@tohoku.ac.jp)
[2]Artificial Intelligence R&D Dept., Information Technology R&D Center, Mitsubishi Electric Corporation, Ofuna, Kamakura, 247-8501, Japan (Nammoto.Takashi@ds.MitsubishiElectric.co.jp; Kato.Ryota @ bp.MitsubishiElectric.co.jp)

loss [4]. Therefore, while generating 3D models from a single image is a promising method to supplement the lack of prior information, methods are needed to appropriately handle the shape uncertainty in generative models and utilize it for control.

B. Visual Feedback Control in Robotic Manipulation

Visual Servoing (VS) is a method for controlling robots while reducing uncertainty in 3D models. [4]. Two representative VS methods are Position-Based Visual Servoing (PBVS) and Image-Based Visual Servoing (IBVS), each possessing different characteristics. [5]

PBVS is a method that directly minimizes spatial pose error using camera pose and a 3D model of the object, demonstrating high convergence even with large initial pose errors. However, it has high sensitivity to 3D model accuracy and camera calibration errors, requiring careful planning for real-world applications [6].

In contrast, IBVS directly uses the misalignment of features in the image as the reference , offering the advantage of being more robust to the object's geometric model and calibration accuracy. However, when using a single camera, the field of view is significantly constrained. In situations with large initial position errors, the object may drift out of the field of view, potentially causing control failure [7].

Hybrid Visual Servoing has been proposed as a method that complements the characteristics of both approaches. This method has been shown to be effective. This technique utilizes PBVS for rough positioning near the target, followed by IBVS for high-precision fine adjustment [4]. This approach achieves high-precision pose alignment while controlling errors resulting from incomplete 3D models and camera calibration errors.

Therefore, hybrid VS remains an effective method for robust robot control even in environments with limited model accuracy.

C. Positioning and Contributions

The objective of this research is to achieve high-precision robot control under shape uncertainty by addressing two key aspects: "shape uncertainty in 3D generative models" and "visual servo control." Fig.1 shows an overview diagram of the proposed approach. Specifically, a pre-trained large-scale 3D generative model is utilized as a distribution of shape uncertainty aligned with the input image. The selection of the optimal shape candidate is then made based on two metrics: geometric error and visual similarity.

It is intuitive to understand that increasing the number of shape candidates improves the probability of including a "plausible" model that aligns with the observed image. However, given that even state-of-the-art techniques require at least 20 seconds to generate a single model [8], pregeneration and storage of a large number of shapes are unrealistic for the real-time robot control envisioned in this research. In order to address the trade-off between accuracy and computational cost, this study has two primary objectives. Firstly, it seeks to reduce the number of candidate shapes. Secondly, it explores the minimum number required for high-precision control. The integration of this model selection mechanism into a visual servo control system enables high-precision pose alignment, effectively reducing shape ambiguity on the object's back side caused by the single-camera viewpoint.

II. Related Work

A. Integration of Generative Models and Visual Servoing

VS requires preparation of the target image and assumes overlap between the initial and target fields of view. Pathre et al. *Imagine2Servo* enabled large-disparity navigation by generating midpoint and target views using generative models, then applying IBVS stepwise with these as goals [9]. This system has achieved long-distance movement and object manipulation with real robotic platforms in both single-eye and multi-eye settings. This approach has proven to be effective in scenarios where target images cannot be prepared. The integration of generative models and VS has effectively addressed challenges such as 'no target video', 'wide-baseline navigation' and 'feature point loss'.

B. 3D Reconstruction from Sparse Views Using Generative Models and Their Applications to Robotics

Recent research has actively explored estimating 3D shapes from 1 to 5 images using generative models. *Zero-1-to-3* by Liu et al. achieved significantly higher accuracy than existing methods by combining multi-view image images using a viewpoint-conditional generative model and reconstructing the object's 3D shape from them [10]. This method is characterized by not requiring prior CAD data and can be applied to various objects.

Generative models have also been applied to the generation of multimodel grasp points. Jiang et al. presented a method for probabilistically generating 6-DoF grasp candidates [11]. Additionally, Huang et al. improved accuracy by generating grasp points based on textual instructions [12]. Both approaches do not require prior CAD data and are useful when prior measurements or multi-view imaging is difficult.

III. Proposed Method

In visual servoing, it is difficult to obtain accurate 3D models of objects or images of the target viewpoints in advance. In such situations, information about the object's back side or blind spots is insufficient, making high-precision alignment difficult with conventional VS methods.

This research focuses on guiding a robot camera to the back view of an object using only a single RGB image as initial input, ultimately achieving target alignment. However, no teacher information, such as target images or complete 3D models, is available. Therefore, it is necessary to assume the unknown shape based on the view of the visible region,and alignment must be achieved by sequentially adjusting the pose.

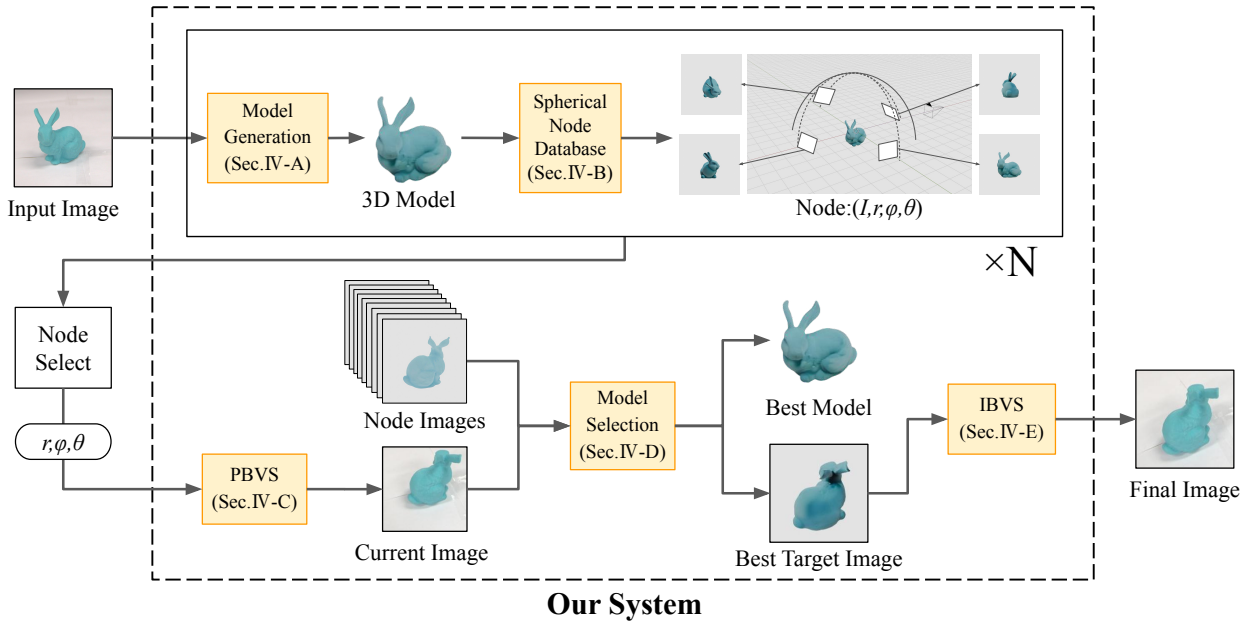


Fig. 2: Overall architecture of the proposed system

The general configuration of the visual servo system proposed in this study for control tasks under such conditions of insufficient information and high uncertainty is shown in Figure 2.

This system is designed to operate robustly even in situations where prior measurement or multi-view imaging is difficult. Specifically, even when only a single initial image is available, it actively controls the camera while handling shape uncertainty , guiding it to the target viewpoint to achieve precise alignment. The system consists of five modules that work together to control operations sequentially.

A. Initial Model Generation Module

3D models generated from images obtained from a single viewpoint are not uniquely determined, and there are numerous candidates.

This research formulates this shape indeterminacy as an uncertainty distribution. Since manually creating a rule-based distribution description is impractical, we adopt a pre-trained large-scale 3D generative model [8] and utilize it as a shape distribution aligned with the input image. Analytically describing the high-dimensional distribution defined by the generative model is difficult. Therefore, we used a Monte Carlo method to obtain multiple samples and estimate the uncertainty of the shape in the blind spot based on the diverse set of hypothesis shapes of objects.

Given an initial image I_0, we perform one-shot 3D shape generation using the generative model $\mathcal{D}$. Specifically, we generate a set of candidate meshes $\{M_i\}_{i=1}^{N}$ by varying the random seed used in the sampling process. In our implementation, we use $N = 10$ samples, forming a diverse set of hypothesis shapes of objects:

$$M_i = \mathcal{D}\big(I_0; \texttt{seed} = i\big), \quad i = 1, \ldots, N. \tag{1}$$

B. Spherical Node Database Module

For each candidate mesh M_i $(i = 1, \ldots, N)$, we define a set of points of view, known as nodes, using a spherical coordinate grid $\mathbf{t}_{k,m,n} = (r_k, \theta_n, \phi_m)$. A virtual camera is placed at each node to render the 3D model's images from that viewpoint.

The position $\mathbf{p}_{k,m,n} \in \mathbb{R}^3$ and the orientation $R_{k,m,n} \in SO(3)$ of the virtual camera at node (k, m, n) are defined as follows:

$$\mathbf{p}_{k,m,n} = r_k \begin{bmatrix} \sin\phi_m \cos\theta_n \\ \sin\phi_m \sin\theta_n \\ \cos\phi_m \end{bmatrix}, \tag{2}$$

$$\begin{aligned} \mathbf{z}_{k,m,n} &= -\frac{\mathbf{p}_{k,m,n}}{\|\mathbf{p}_{k,m,n}\|}, \\ \mathbf{y}_{k,m,n} = \mathbf{z}_{k,m,n} \times \mathbf{u}, \quad \mathbf{x}_{k,m,n} &= \mathbf{y}_{k,m,n} \times \mathbf{z}_{k,m,n}, \\ R_{k,m,n} &= [\,\mathbf{x}_{k,m,n}\ \mathbf{y}_{k,m,n}\ \mathbf{z}_{k,m,n}\,], \end{aligned} \tag{3}$$

where $\mathbf{u} = [0, 0, 1]^\top$ is the ascending canonical vector.

Each mesh M_i is rendered from the viewpoint $\mathbf{p}_{k,m,n}$, and the resulting RGB image $I_{i,k,m,n}$ is stored.

Here, we define $\mathcal{R} \times \Phi \times \Theta$ as a three-dimensional grid composed of the discrete set of hemispheres $\mathcal{R} = \{r_k\}$, the set of latitude angles $\Phi = \{\phi_m\}$, and the set of longitude angles $\Theta = \{\theta_n\}$, each discretized at constant degrees. Since all candidate models share the same spherical grid $\mathcal{R} \times \Phi \times \Theta$, subsequent PBVS operations can refer to the target points of view independently of any specific model.

C. PBVS Control Module

This module performs coarse alignment by moving the camera toward the target viewpoint (that is, the target node). To ensure model independence, the target pose is defined in a shared node coordinate system using the parameters introduced in equations (2) and (3). The target transformation matrix is given by:

$$T_{k,m,n} = \begin{bmatrix} R_{k,m,n} & \mathbf{p}_{k,m,n} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{4}$$

Let T_C denote the current camera pose, and T_G the desired pose (target node). The relative transformation between the two is computed as follows:

$$T_e = T_G T_C^{-1} = \begin{bmatrix} R_e & \mathbf{t}_e \\ 0 & 1 \end{bmatrix}, \tag{5}$$

and used to generate a position-based feedback command for the robot arm or camera mount.

Since all candidate models share the same spherical node grid, PBVS can be applied independently of the shape parameters of the object. Additionally, the generated meshes are placed in real-world scale, eliminating the need for additional scale calibration. This configuration ensures reliable convergence to the target node, even in scenarios with large viewpoint disparities.

D. Model Selection Module

Once the camera reaches the neighborhood of the target node at time $t^\star$, we evaluate the similarity between the real observation $I_{t^\star}$ and the set of rendered images $\mathcal{R}_{k,m,n} = \{I_{i,k,m,n}\}_{i=1}^{N}$ corresponding to each candidate mesh M_i. The best matching model M^* is then selected based on two complementary criteria: geometric alignment and visual-semantic similarity.

Geometric alignment is measured by the Chamfer distance between edge maps, which quantifies the average nearest-neighbor distance between the observed and rendered object edges.

Visual-semantic similarity is evaluated by cosine similarity between the features of the CLIP encoded image [13], providing a semantic level comparison.

1) Feature Extraction:

- Edge map: An edge map $E(I) = \text{Canny}(I) \in \{0, 255\}^{L\times L}$ is computed using the Canny edge detector.
- CLIP embedding: A ViT-L/14 encoder ϕ_{CLIP} is used to obtain a normalized feature vector:

$$\mathbf{f}_{\text{CLIP}}(I) = \frac{\phi_{\text{CLIP}}(I)}{\|\phi_{\text{CLIP}}(I)\|_2} \in \mathbb{R}^{768}. \tag{6}$$

2) Similarity Metrics:

- Chamfer Distance: For two edge maps E_1 and E_2, the Chamfer distance is defined as

$$D_{\text{Chamfer}}(E_1, E_2) = \tfrac{1}{2}\Big(\tfrac{1}{|E_1|}\sum_{x\in E_1} \text{DT}_{E_2}(x) + \tfrac{1}{|E_2|}\sum_{x\in E_2} \text{DT}_{E_1}(x)\Big), \tag{7}$$

  where $\text{DT}_E(x)$ denotes the Euclidean distance from pixel x to the closest pixel in the edge set E [14].
- CLIP Similarity:

$$S_{\text{CLIP}}(I_{t^\star}, I_{i,k,m,n}) = \mathbf{f}_{\text{CLIP}}(I_{t^\star})^\top \mathbf{f}_{\text{CLIP}}(I_{i,k,m,n}) \in [0,1]. \tag{8}$$

3) Score Normalization and Fusion: For each candidate, we compute the following.

$$\begin{aligned} d_i &= D_{\text{Chamfer}}\left(E(I_{t^\star}), E(I_{i,k,m,n})\right), \\ s_i &= S_{\text{CLIP}}(I_{t^\star}, I_{i,k,m,n}). \end{aligned} \tag{9}$$

These scores are normalized to the range $[0, 1]$ as:

$$\begin{aligned} \hat{d}_i &= \frac{d_i - \min_j d_j}{\max_j d_j - \min_j d_j + \varepsilon}, \\ \hat{s}_i &= \frac{s_i - \min_j s_j}{\max_j s_j - \min_j s_j + \varepsilon}, \end{aligned} \tag{10}$$

where $\varepsilon = 10^{-6}$ is a small constant for numerical stability.

The final fusion score is defined as

$$\begin{aligned} F_i &= w(1 - \hat{s}_i) + (1 - w)\hat{d}_i, \\ w &\in [0, 1], \end{aligned} \tag{11}$$

where w is a hyperparameter that controls the relative importance of semantic similarity (CLIP) and geometric consistency (Chamfer). In this study, we set $w = 0.5$.

4) Optimal Model Selection: Finally, the model with the lowest fusion score is selected as the best model:

$$i^* = \arg\min_i F_i. \tag{12}$$

This selection enables the 3D shape of the object, which was unclear based on the initial image , to be selected as the best solution that matches more closely the real environment.

E. IBVS Control Module

Once the best model M^* is selected, the alignment control for slight misalignment is performed using that model. In IBVS, the error between the RGB values observed on the image and the corresponding image features rendered from the model is directly feedback-controlled.

A Region of Interest (ROI) is defined in the center of the image. Let P denote the number of pixels in the ROI, and let each pixel's RGB values be defined as $[R_p, G_p, B_p]^\top$. The vector of image characteristics $\mathbf{s}(I)$ is then defined as:

$$\begin{aligned} \mathbf{s}(I) &= \left[R_1, \ldots, R_P,\ G_1, \ldots, G_P,\ B_1, \ldots, B_P\right]^\top \\ &\in \mathbb{R}^{3P}. \end{aligned} \tag{13}$$

Given a reference image $\hat{I}$ rendered from the best model M^* at the pose of the target node, the image error is computed as:

$$\mathbf{e}_I = \mathbf{s}(I) - \mathbf{s}(\hat{I}). \tag{14}$$

Since global alignment of the camera pose is already achieved at the end of PBVS, the remaining errors that IBVS must converge to are primarily limited to translation, scaling, and rotation of the optical axis misalignments.

For each control parameter $\boldsymbol{p} = [\Delta x,\ \Delta y,\ \Delta z,\ \Delta\alpha]^\top \in \mathbb{R}^4$ (translation, scale, rotation), generate four difference images by adding a small amount ε to each component:

$$I^+_{(j)} = \text{Render}\big(M^*, \boldsymbol{p} = \varepsilon \mathbf{e}_j\big).$$

The image Jacobian $J \in \mathbb{R}^{3P\times 4}$ is constructed as:

$$\begin{aligned} J &= [J_1,\ J_2,\ J_3,\ J_4], \\ J_j &= \frac{\mathbf{s}(I^+_{(j)}) - \mathbf{s}(\hat{I})}{\varepsilon} \in \mathbb{R}^{3P}. \end{aligned} \tag{15}$$

The control command is computed using the pseudoinverse of the Jacobian:

$$J^\dagger = (J^\top J)^{-1} J^\top,$$

and the optimal update is given by:

$$\boldsymbol{p} = -\lambda\, J^\dagger\, \mathbf{e}_I, \qquad 0 < \lambda \leq 1, \tag{16}$$

where λ is a gain parameter. The resulting $\boldsymbol{p}$ is applied directly as the control input to the robot end effector:

$$\boldsymbol{\xi} = [\Delta x,\ \Delta y,\ \Delta z,\ \Delta\alpha]^\top.$$

Under these conditions, IBVS also shows accurate local convergence, enabling high-precision alignment within a small range of error. Ultimately, the IBVS control guides the camera to a pixel-level aligned position to the target image, achieving the overall control objective of the system.

IV. Experiments

A. Objective

The purpose of the experiment is to quantitatively evaluate the effectiveness of the proposed method in real world settings. We focus on two main aspects: (i) the effect of the number of candidate shapes on the convergence success rate, and (ii) the effect of model selection strategy and HVS on final alignment accuracy. From this, we identify the effectiveness of our approach in achieving the best model selection and stepwise visual control while preserving shape uncertainty.

B. Experimental Setup

For real-world implementation, the visual servo system was built using a ToroboArm Mini mounted with a Basler acA4112-30uc RGB camera (resolution: 1000×1000), as shown in Fig. 3. The camera was mounted in an overhead configuration.

The camera field of view is designed to fully project the area near the end effector. Furthermore, the size of each target object is within 10 cm cubed.

To test the diversity of shape hypotheses and the robustness of the proposed staged control strategy, we used six different types of physical objects, illustrated in the first and second rows of Fig. 4:

- Bunny: a moderately dense model with a complex outline.

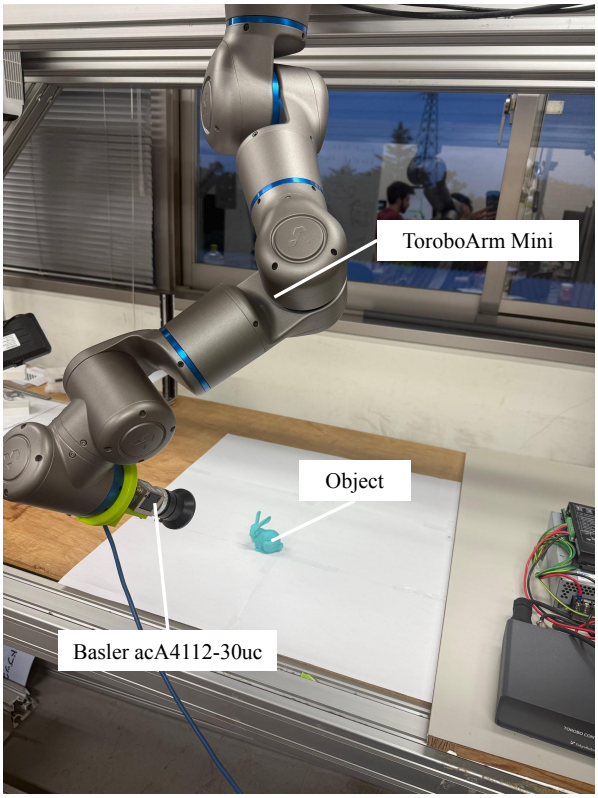


Fig. 3: Experimental setup: ToroboArm Mini, camera, and object placement

- Dragon: a complex model with extreme detail and waviness.
- Hex Case: a hybrid structure featuring both hollow and closed surfaces.
- Cone: a combination of sharp tips and smooth curves.
- Stone1 and Stone2: irregular and rough shape.

For each object, a single view image was captured and used as input to the Hunyuan3D-2.0 model [8], which generated nine candidate 3D models. For each mesh M_i $(i = 1, \ldots, 9)$, we define a set of candidate viewpoints in spherical coordinates, discretized over radius r, elevation ϕ, and azimuth θ as: For each object, acquire one initial image and generate 1 to 9 candidate models using Hunyuan3D-2.0 [8]. For each model M_i $(i = 1, \ldots, 9)$, introduce spherical coordinates and discretize the set of candidate points of view using radius r, latitude ϕ, and longitude θ as follows:

$$\begin{aligned} \mathcal{R} &= \{r_k = 0.10k \mid k = 1, \ldots, K\}, \\ \Phi &= \{\phi_m = 10m^\circ \mid m = 0, \ldots, 9\}, \\ \Theta &= \{\theta_n = 10n^\circ \mid n = 0, \ldots, 35\}, \\ K &= \left\lfloor \frac{r_{\max}}{0.10} \right\rfloor, \end{aligned} \tag{17}$$

We set $r_{\max} = 0.4$ m and define a spherical viewpoint within the movable range of the robot. We render using the blenderproc.camera package based on the position and orientation derived from equations (2) and (3).

(a) Bunny (b) Dragon (c) Hex case (d) Cone (e) Rock1 (f) Rock2

Fig. 4: Six physical objects used in the experiments and their rendered backside models. (Top: front view; second row: back view; third row: successful model; bottom: failed model)

To create a sufficiently different viewing environment from the initial viewpoint, the target viewpoint for PBVS was set to spherical coordinates (r, ϕ, θ). We set $r = 0.3m$, $\phi = 45°$, and randomly selected θ within the range $+160° \sim +200°$ from the initial viewpoint.

During IBVS, an ROI 200×200 was set in the image center to reduce the calculation cost and avoid miscontrol due to background.

C. Evaluation Metrics

To comprehensively evaluate the performance of the proposed method, we design two complementary metrics: (i) convergence stability in visual servoing control and (ii) visual consistency between the captured and target images.

1) Convergence Success Rate: We change the number of candidate models and measure the convergence success rate as a function of this number. Convergence is considered successful if the following two conditions are simultaneously satisfied at the end of IBVS:

- The norm of the estimated vector of the control parameters $\hat{\boldsymbol{\xi}}$ falls below a predefined threshold $\|\hat{\boldsymbol{\xi}}\| < \varepsilon, \varepsilon = 0.001$.
- A human observer confirms that the object's position and orientation in the final camera image visually match those in the target image (visual inspection).

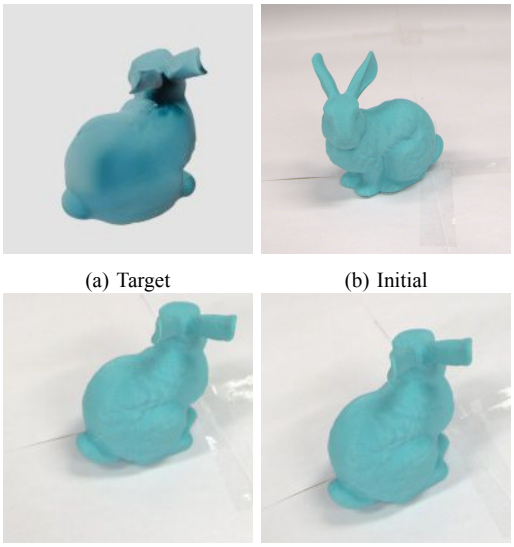
Here, ε is an experimentally determined threshold selected because no clear improvement in robot behavior was observed compared to lower values after control convergence. Introducing both a numerical convergence condition and visual consistency achieves both the rejection of misconvergence and the visualization of realistic convergence success.

2) Image Alignment Accuracy: To assess the effects of model selection and hybrid VS, we evaluated the similarity between the target image and the captured images at three stages: the initial view, after coarse alignment (PBVS) and after fine alignment (IBVS). An example of a Bunny object is shown in Fig. 5.

We use the following quantitative metrics:

- Chamfer Distance: Canny edge detection is applied to extract contours from each image. The average of the bidirectional nearest-neighbor distances between the two sets of edge points is computed to quantify the geometric consistency (unit: pixels).
- Mean Squared Error (MSE): In the image, a 200×200 square ROI is defined centered on the object. The MSE of the RGB values is then computed between the captured and target images within this ROI.

The Chamfer distance reflects shape alignment accuracy (in terms of contour matching), while the MSE evaluates pixel-level visual fidelity, including brightness and color



(a) Target (b) Initial

(c) After PBVS (d) After IBVS

Fig. 5: Target and captured images at each alignment stage (Bunny object)

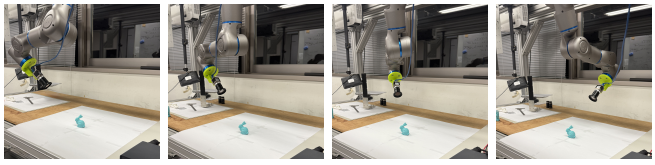


Fig. 6: Motion sequence of the robotic arm

consistency. Together, these two metrics provide a comprehensive assessment of geometric and photometric consistency. Since the generated shape candidates preserve texture information, pixel-based evaluations incorporating color are considered appropriate.

The ROI size of 200×200 pixels was chosen based on the average occupied area of the object in the camera view. It is designed to match the scale and resolution of the experimental setup, considering the camera field of view, resolution, and object size.

D. Results and Discussion

We implemented the proposed method on a real robotic platform. Figure 6 shows the sequence of motion of the robotic arm that is convergent under IBVS control.

1) Convergence Success Rate: We varied the number of candidate shapes per object $(1, 3, 5, 7, 9)$ and evaluated the resulting convergence performance and the final alignment accuracy. Figure 7 shows the convergence success rate in different candidate counts.

Using a single candidate model (i.e., without model selection) led to a significant drop in success rate, especially for symmetric or irregular objects. Figures 4 (rows 3 and 4) show back views of the physical objects and the selected models in both successful and failed trials.

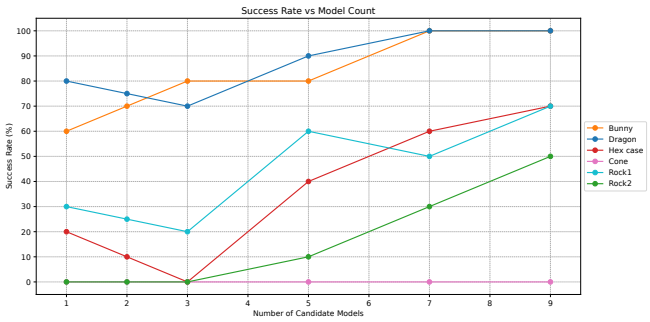


Fig. 7: Convergence success rate vs number of candidate models for each object

TABLE I: Chamfer distances and MSE for each object

Object	Chamfer Distance (px)			MSE [$\times 10^{-3}$]		
	Initial	After PBVS	After IBVS	Initial	After PBVS	After IBVS
Bunny	45.87	41.12	17.87	2.17	2.01	0.76
Dragon	56.30	47.31	13.20	1.20	1.30	0.58
Hex case	60.29	35.43	10.24	0.71	0.98	0.44
Cone	–	–	–	–	–	–
Rock 1	26.45	21.94	16.79	1.29	2.01	0.96
Rock 2	63.92	56.33	49.08	1.28	1.13	1.01

For highly symmetric or irregular objects such as Cone and Rock 2, increasing the candidate count did not always resolve the failure. In particular, the cone failed in all trials, which did not yield alignment metrics. This is attributed to its indistinct visual features on the back side, rendering model selection ineffective under such uncertainty.

In contrast, for all other objects, the success rate improved consistently with more candidates, validating the proposed model-switching strategy. This shows that the approach of using seed diversification to represent shape uncertainty in one-shot generation and selecting the model most consistent with observations is effective in avoiding global geometric mismatches. In particular, Bunny and Dragon, despite limited texture, achieved success 100% due to their distinctive geometry.

These results demonstrate that complementing single-shot generation with model diversity significantly enhances convergence robustness.

Although the success rate continued to improve to up to nine candidates, it did not saturate, indicating the potential for further gains. However, increasing the number of candidates increases the generation time and storage cost, requiring a practical balance between performance and efficiency.

2) Image Alignment Accuracy: Table I summarizes the Chamfer distances and MSE (scaled by 10^{-3}) for each object.

Both Chamfer distance and MSE improved after PBVS, with further refinement by IBVS. For Cone, convergence was not achieved, so no data are reported.

For Bunny, the Chamfer distance decreased from 45.87 px to 17.87 px and the MSE from 2.17 to 0.76, an approximate 65% reduction, demonstrating a significant improvement.

For some objects (Dragon, Hex case, Rock1), a temporary

increase in differences was observed after PBVS. However, consistent recovery of alignment quality was ultimately achieved through IBVS. This indicates that large parallax alignment prioritized moving objects to their target positions, temporarily losing pixel-level matching. Subsequent IBVS then achieved the final alignment.

These results confirm the effectiveness of our staged strategy: PBVS achieves coarse global alignment, while IBVS ensures fine pixel-level adjustment. The method proves robust and accurate across diverse geometries, even under shape uncertainty.

V. Conclusion

This study proposed an integrated visual servoing (VS) framework that performs staged alignment - from coarse to fine - while retaining shape uncertainty in multiple 3D hypotheses generated from a single RGB image using a diffusion-based generative model.

Real-world experiments conducted on various objects evaluated both the convergence success rate and shape alignment accuracy. The results demonstrated that even with fewer than ten candidate shapes, the proposed model selection mechanism significantly improved the robustness of alignment. In particular, objects with well-defined geometric features achieved high-precision and stable visual alignment.

However, for objects with self-symmetrical shapes or lacking visual features, increasing the number of candidates still tends to cause misconvergence. Improving this through highly discriminative visual features (e.g. DINO or semantic features) or integrating observations from multiple viewpoints remains a future work.

Furthermore, since current experiments were conducted in a simplified environment with a white background, future efforts should consider visual features that are invariant to background conditions and explore the use of multimodal information for enhanced robustness.

Based on these results and challenges, this method is positioned as an effective framework for visual servo control that enables approaching and aligning with objects using only single image information, even in unknown environments, without the need for pre-prepared target images or exact 3D models as in previous approaches.

References

[1] Guillaume Walck and Michel Drouin. Progressive 3d reconstruction of unknown objects using one eye-in-hand camera. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 971–976, 2009.

[2] Bo Li, Xiaolin Wei, Fengwei Chen, and Bin Liu. 3d colored shape reconstruction from a single rgb image through diffusion, 2023.

[3] Chenru Jiang, Chengrui Zhang, Xi Yang, Jie Sun, Yifei Zhang, Bin Dong, and Kaizhu Huang. Consistency diffusion models for single-image 3d reconstruction with priors, 2025.

[4] You-Fu Chiang, Yen-Heng Liu, and Chun-Ta Chen*. Hybrid visual servo control for point-to-point localization of an autonomous wheeled mobile robot, 2022.

[5] Seth Hutchinson, {Gregory D.} Hager, and {Peter I.} Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, pp. 651–670, 1996.

[6] Benoit Thuilot, Philippe Martinet, Lionel Cordesses, and Jean Gallice. Position based visual servoing: Keeping the object in the field of vision. Vol. 2, pp. 1624 – 1629 vol.2, 02 2002.

[7] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, pp. 313–326, 1992.

[8] Zibo Zhao et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025.

[9] Pathre et al. Imagine2servo: Intelligent visual servoing with diffusion-driven goal generation for robotic tasks. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13466–13472, 2024.

[10] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023.

[11] Rosa Wolf, Yitian Shi, Sheng Liu, and Rania Rayyes. Diffusion models for robotic manipulation: A survey, 2025.

[12] Toan Nguyen, Minh Nhat Vu, Baoru Huang, An Vuong, Quan Vuong, Ngan Le, Thieu Vo, and Anh Nguyen. Language-driven 6-dof grasp detection using negative prompt guidance, 2024.

[13] Radford et al. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021.

[14] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, Vol. 8, No. 19, pp. 415–428, 2012.