IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Visual Servoing-Based Active Vision for 3D Object Reconstruction

Ekrem Misimi[1], Sverre Herland[1], and François Chaumette[2]

***Abstract*—In this letter, we present a novel dual-task, closed-loop, visual servoing-based active vision framework in an eye-in-hand configuration. The proposed active vision framework continuously drives the camera motion by coupling continuous Next-Best-View (NBV) planning and visual servo control within a unified formulation, is NBV-objective-agnostic, and enables real-time, closed-loop exploration of objects. We demonstrate how this approach can be applied to the 3D reconstruction of static volumetric objects. The approach is validated in the real world with a diverse set of relevant objects and we observe that the visual servo scheme produces smooth exploration trajectories that keep the camera focused at the object. We also show that our gradient-based continuous NBV-strategy is highly competitive with baseline strategies that leverage global viewpoint sampling and results in efficient exploration with strong object coverage.**

***Index Terms*—Visual Servoing, Reactive and Sensor-Based Planning, Planning under Uncertainty, RGB-D Perception**

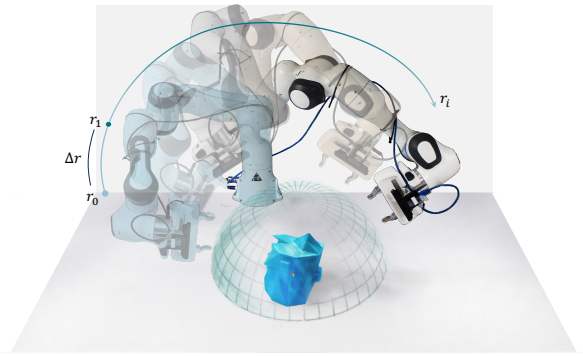

Fig. 1. Camera motion is controlled through two coupled visual servoing tasks. The *primary task* keeps the object centered in the image while the *secondary task* dynamically controls camera motion along a hemisphere, guided by an NBV objective.

I. Introduction

ONE of the main challenges in robot vision and robotic manipulation of objects is capturing and recovering the 3D shape of an unknown object by means of available visual data. The most naive and intuitive way using a robot arm and eye-in-hand camera is to scan the object from all sides and construct an aligned, high-resolution accumulated point cloud [1]. However, such an approach based on predetermined trajectories is inefficient since it gathers redundant information. Therefore, the need for more efficient and autonomous strategies is high [2]. Humans utilize active visual perception to peek at occluded or novel sides of an object during visual examination. In the same way, active vision consists of a strategy to control the motion of a camera to explore and model the environment [3], [4]. In the context of 3D reconstruction [2], [5], its purpose is to explore new or hidden sides of an unknown object, by iteratively changing the camera pose. A central challenge in active vision remains the lack of integrated control strategies that not only select informative viewpoints, but also autonomously reach them in a closed-loop manner as most existing Next Best View (NBV) approaches decouple planning from control, leading to discrete execution. Therefore, coupling viewpoint selection with a task that dynamically controls the camera toward high-information regions remains an important challenge.

Manuscript received: August, 27, 2025; Revised November, 3, 2025; Accepted November, 26, 2025.

This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the BIFROST project (FRIPRO RCN 313870).

[1]Ekrem Misimi and Sverre Herland are with SINTEF Ocean, Norway ekrem.misimi@sintef.no

[2] François Chaumette is with Inria, Univ Rennes, CNRS, IRISA - Rennes, France. francois.chaumette@inria.fr

Digital Object Identifier (DOI): see top of this page.

Viewpoint selection during exploration is a crucial step [2], [6]. There are numerous approaches to this planning problem [5] — a common one being NBV [7]. NBV is a greedy search strategy that selects the view expected to rapidly reveal the most new information. This process is repeated until no further sufficiently informative views can be obtained.

Another important aspect is the use of prior knowledge. Model-based approaches exploit partial or full object models, while model-free methods [8] operate without priors, often using volumetric representations [8]–[11] to compute metrics such as ray casting or information gain [10]–[12]. In practice, most approaches incorporate some form of prior models, maps [8], [13], or datasets for training learned NBV planners [2], [9], [14]. These trained models guide exploration by distilling prior knowledge into informed viewpoint selection. While prior knowledge and data about the object may result in more efficient viewpoint selection, these methods tend to focus on specific classes or distributions of objects. This work is predominantly concerned with agnostic methods that work with arbitrary objects.

In most works on active vision [2], [8], [9], [11], [15], [16], the objects are placed on a desktop platform so the visual sensor cannot explore the bottom of the object and the exploration is done with a movable visual sensor attached to a robot arm. Alternative techniques, where the robot grasps and lifts the object toward a static camera for exploration and reconstruction, have also been reported [17]. Previous approaches are limited to simulation only [8], [18]–[22], rely on discrete view candidates [23], or depend on a bounding box [15]. Two recent approaches [15], [16] are promising, but their NBV planning is aimed at revealing occluded regions to improve grasp stability rather than enabling full 3D reconstruction. Morrison et al. [16] propose a multi-view grasp controller that selects informative viewpoints to reduce the entropy of

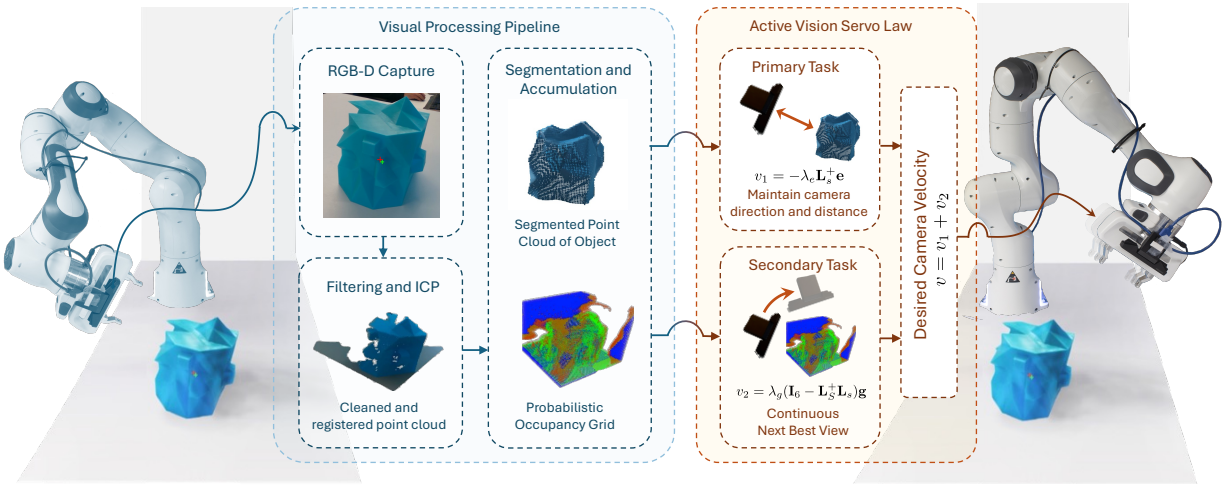

Fig. 2. Overview of proposed active vision framework. A visual processing pipeline captures, cleans, registers, and accumulates sensor data from an eye-in-hand RGB-D camera. The segmented point cloud and a probabilistic occupancy map are used as input to a dual-task visual control law that keeps the camera trained on the object while exploring through gradient-based maximization of an NBV objective.

grasp quality predictions. Breyer et al. [15] propose a closed-loop NBV planner that integrates visual measurements into a voxel map to reveal occluded regions and replan grasps, to improve grasp stability.

Virtually all of the aforementioned NBV strategies can be characterized as generate-and-test approaches based on global sampling. This means that a discrete set of candidate poses are generated, scored by an NBV objective function, and then the best one is selected as a navigation target. While these strategies guarantee that all camera poses have a chance to be considered and are shown to work well in many cases, they have trade-offs, including the computational burden of evaluating a large set of candidates and, as a corollary, the inability to continuously and quickly react to new information that arrives while moving. Additionally, they often necessitate some form of prior knowledge about the object or global search space in order to generate valid candidates and navigate. Therefore, developing efficient continuous and closed-loop solutions to the NBV-problem that only rely on local information is an important direction of research, as they embody reactive control, where what the camera sees instantly determines where the robot moves next.

A closely related recent work that presents a fast local approach to NBV is given in [21]. The authors construct an information-oriented NBV objective that is differentiable with respect to the 3D position and 2D viewing direction of the camera (5-DoF). This allows a gradient-based control law to be derived that only depends on a single evaluation of the objective function and its analytic derivatives. The approach is shown to produce exploration trajectories that yield efficient object coverage, although only in a restricted simulation benchmark involving scanning of tomato plants.

Our work shares the high-level motivation and goals of the method presented in [21], but with some key differences. Instead of fully controlling the camera with a 5-DoF NBV-gradient, we present an approach based on visual servoing that splits the problem into two tasks. The primary task is responsible for pointing the camera at the object and maintaining a fixed distance, which ensures a smooth scan, keeps the object in focus, and maintains a suitable distance between the object and the depth sensor, which is essential for good depth readings for many real-world RGB-D cameras. Meanwhile, the secondary task performs active vision in the null-space of the former, effectively reducing the dimensionality of the exploration problem to traversal of a 2D surface manifold. This simplifies the exploration problem and allows us to cheaply estimate the NBV gradient with finite differences (4 samples) for any NBV objective function - differentiable or not. In contrast to the simulation benchmark in [21], we also demonstrate feasibility on a physical robot with a wider range of test objects with varied shape, size and geometry.

To the best of our knowledge, no existing framework achieves real-time, closed-loop active vision that simultaneously couples continuous NBV-planning and 3D reconstruction through a dual-task visual servoing formulation on a physical robotic platform. In particular, most lack an autonomous visual servoing strategy that tightly integrates NBV planning with camera gaze motion control, a key aspect to enable continuous and seamless exploration. Therefore, we present a visual servoing-based approach to active vision that does not rely on any prior knowledge, or information on the geometry of the objects it is exploring, that takes care of both NBV-planning and 3D object reconstruction in a continuous way and in real time.

Our contributions are summarized as follows:

- We present a general-purpose, real-time framework for active vision based on dual-task visual servoing, where exploration and viewpoint selection are integrated within a unified control law to achieve smooth, continuous camera motion in closed loop.
- The approach is NBV-objective-agnostic, relies only on local visual information, and achieves computationally efficient and smooth camera motion through reactive control.
- We perform real-world evaluations on a physical robot with a diverse set of objects and compare to baselines

that leverage global NBV goal sampling.
- We show that despite only controlling based on local information, our gradient-based NBV-strategy produces exploration trajectories with strong object coverage that are highly competitive with global sampling strategies that leverage prior knowledge about the search space.

II. Methodology

An overview of our proposed active vision framework is shown in Figs. 1 and 2. It consists of a point cloud processing pipeline that aggregates and refines sensor data, and a visual control law that generates desired camera velocities.

A. Point Cloud Processing

During operation, the system continuously filters, registers, accumulates, and segments point clouds captured from a wrist-mounted RGB-D camera. Except otherwise stated, the algorithms used are implemented with the PCL Library [24].

1) Filtering: The raw point cloud is simplified to 2.5 mm resolution with a voxel-based filter and cropped to a box centered on the camera frustum ($x, y \pm 15$ cm, $z \in (5, 40)$ cm. We also apply a statistical outlier rejection filter to remove spurious points caused by noisy depth readings. Residual noise in the centroid estimation is mitigated through a low-pass filter, ensuring stable convergence of the primary task.

2) Registration: The cleaned point clouds are registered into a canonical coordinate frame, which we set to be the robot base link frame. We use the Point-to-Plane Iterative Closest Point (ICP) algorithm [25] with surface normals estimated with a search radius of 5 mm. The ICP registration is configured with a transformation and fitness epsilon of 10^{-8}, and a maximum of 20 iterations. Forward kinematics from the robot is used to form a strong initial guess of the solution pose, and we use a distance-based correspondence rejector of 2 mm to regularize the final aligned pose. After each registration, we concatenate the newly aligned points with the target points and simplify with another 2.5 mm voxel filter to form a target for the next iteration.

3) Accumulation: The registered point clouds are also accumulated in a dense, probabilistic occupancy voxel grid. We use the Octomap package [26] with a voxel resolution of 5 mm, which allows us to keep track of not just which parts of the scenes that are occupied, but also which parts that are free and which parts we are still unsure about. This method enables robust accumulation of evidence from multiple camera viewpoints, and probabilistic modeling of occupancy is essential for the information-oriented active vision objective outlined in Section II-B.

4) Segmentation: The segmentation pipeline recovers the object of interest from the accumulated occupancy grid. First we threshold the volumetric occupancy probabilities to recover a sparse point cloud. Then we identify and remove the ground plane with RANSAC [27] and lastly, we apply Euclidean clustering to extract the largest contiguous cluster, which corresponds to the object of interest. We also compute the bounding box of the object - the Region of Interest (*ROI*) - and sample its occupancy probabilities to track overall entropy (see termination criterion in Section II-D).

B. Visual Servoing Control Law

The camera exploration trajectories are generated with a visual servoing control scheme for eye-in-hand camera configurations. We rely on a dual-task visual servoing control law where the primary task keeps the camera fixed on the object of interest while the secondary task actively explores it. The primary task (the first term) drives a set of visual features $\mathbf{s}$ towards a desired state $\mathbf{s}^*$ by minimizing error $\mathbf{e} = (\mathbf{s} - \mathbf{s}^*)$, while the secondary task (the second term) generates exploratory trajectories in its null-space:

$$\mathbf{v} = -\lambda_e \mathbf{L}_\mathbf{s}^+ \mathbf{e} + \lambda_g (\mathbf{I}_6 - \mathbf{L}_\mathbf{s}^+ \mathbf{L}_\mathbf{s}) \mathbf{g} \tag{1}$$

Here, $\mathbf{v}$ is the desired camera velocity twist, $\mathbf{L}_\mathbf{s}$ is the interaction matrix that links temporal variations in visual features $\dot{\mathbf{s}}$ to camera velocity $\mathbf{v}$ through $\dot{\mathbf{s}} = \mathbf{L}_\mathbf{s}\mathbf{v}$ and $\mathbf{L}_\mathbf{s}^+$ is the corresponding Moore-Penrose pseudo-inverse matrix, $\mathbf{g}$ is the desired motion for the secondary task, $(\mathbf{I_6} - \mathbf{L}_\mathbf{s}^+ \mathbf{L}_\mathbf{s})$ is a projection operator into the null-space of the primary task, and λ_e, λ_g are gain coefficients that regulate the convergence speed of the tasks, following standard VS-notation and definitions [28].

Active vision starts by activating the primary task (at this stage $\mathbf{g} = \mathbf{0}$) and the secondary task is activated only after the primary task error $\mathbf{e}$ has converged.

1) Primary Visual Servoing Task: The primary task maintains focus on the object of interest during exploration by keeping the centroid of the object in the center of the image plane, and ensuring a constant distance Z^* from the camera to the object. If $\mathbf{P}_c = (X, Y, Z)$ is the centroid of the segmented object in the Cartesian coordinate frame of the camera, then the visual servo law minimizes:

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^* = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ Z^* \end{pmatrix} \tag{2}$$

Here Z^* indicates the desired distance between the camera and the centroid of the object/region of interest, which we set to 25 cm. This yields the following interaction matrix:

$$\mathbf{L}_\mathbf{s} = \begin{bmatrix} -1 & 0 & 0 & 0 & -Z & Y \\ 0 & -1 & 0 & Z & 0 & -X \\ 0 & 0 & -1 & -Y & X & 0 \end{bmatrix} \tag{3}$$

This task ensures that the camera is free to explore the object along a hemispherical surface with the object positioned at its center. For the convergence of the error $\mathbf{e}$, we use a threshold on its norm: when $||\mathbf{e}|| < 0.005$ m, we consider the current visual feature $\mathbf{s}$ to be sufficiently aligned with the desired feature $\mathbf{s}^*$, and that $\mathbf{e}$ has effectively converged. We use an adaptive gain for λ_e, namely $\lambda_0 = 2$, $\lambda_\infty = 0.4$, and $\lambda_0' = 30$, following the formulation in ViSP[1].

2) Secondary Visual Servoing Task: Once the primary task has converged, we activate the secondary visual servoing task which seeks to explore the object. From this point onward, both the primary and secondary tasks are executed simultaneously, following the control law given in (1). Our goal here is to produce a velocity twist $\mathbf{g}$ that controls the

[1] https://visp-doc.inria.fr/doxygen/visp-daily/tutorial-boost-vs.html

camera motion towards unexplored areas. Towards this aim, we assume the existence of a scalar-valued NBV objective function $f : \mathbb{R}^6 \to \mathbb{R}$ that scores the information gain of a candidate camera pose/viewpoint $\mathbf{r} \in \mathbb{R}^6$.

We maximize the objective function f by ascending its gradient within the plane orthogonal to the camera direction, resulting in the following expression for gradient $\mathbf{g}$:

$$\mathbf{g} = \mathbf{g}^{local} = \left(\frac{\partial f(\mathbf{r})}{\partial x}, \frac{\partial f(\mathbf{r})}{\partial y}, 0, 0, 0, 0 \right) \tag{4}$$

Indeed, assuming that the primary task error $\mathbf{e}$ is low, the xy-plane in the camera's coordinate frame should be tangential to the implied hemispherical search space.

We estimate the derivatives of f with central differences. Slightly abusing notation, if $\mathbf{r}$ corresponds to the current camera pose, then $\mathbf{r} + \delta\mathbf{x}$ denote a small translation along the local x-axis, and correspondingly for $\mathbf{r} + \delta\mathbf{y}$ along the local y-axis of the camera frame, giving us:

$$\frac{\partial f(\mathbf{r})}{\partial x} \approx \frac{f(\mathbf{r} + \delta\mathbf{x}) - f(\mathbf{r} - \delta\mathbf{x})}{2\delta x} \tag{5}$$

$$\frac{\partial f(\mathbf{r})}{\partial y} \approx \frac{f(\mathbf{r} + \delta\mathbf{y}) - f(\mathbf{r} - \delta\mathbf{y})}{2\delta y} \tag{6}$$

This ensures that the camera is moved in the direction that maximizes immediate information gain and only requires four evaluations of the objective function f, corresponding to virtual displacements in positive and negative x and y directions, which makes it computationally efficient compared to methods that sample the entire space of possible candidate poses in search of a global solution to the NBV problem. It also requires no information about the set of feasible camera poses beyond those in its immediate neighborhood.

In our experiments, we use $\delta x = \delta y = 5$ cm, which were set based on the overall scale and resolution of the setup and empirical testing. Because the objective value can vary strongly, we rescale it to blend nicely with the primary task by a factor of $\alpha^{local} = 10^{-4}$ and clipping the norm of $\mathbf{g}$ such that $||\mathbf{g}|| \leq 1$, thus making more a source direction than a way to control the speed of the end-effector. We use a flat gain of $\lambda_g = 0.33$ and consider the task converged and terminate whenever $||\mathbf{g}|| < \mathbf{1}$ for more than 80% of the last 200 evaluations of the servo law.

C. Information-Based NBV Objective

A common way to construct objectives for active vision problems is to quantify the volumetric uncertainty of the scene and greedily select the next camera viewpoint that maximizes expected information gain, e.g. by maximizing the entropy of rays cast from a camera pose [12], [21]. We employ a similar approach here, but note that the proposed servo law is agnostic to the choice of NBV objective.

Let $\mathbf{r}$ denote a 6-DoF candidate camera pose consisting of a position with origin $\mathbf{r}_o$ and orientation $\hat{\mathbf{r}}$ with basis vectors $\hat{r}_x, \hat{r}_y, \hat{r}_z$. The image plane is spanned by $\hat{r}_x, \hat{r}_y$ and normal to $\hat{r}_z$. Analogous to the actual depth sensor, our information-oriented objective shoots out rays from $\mathbf{r}_o$ in a pyramid cone centered on $\hat{r}_z$, and then aggregates expected entropy as they pass through the volume representing our current belief of the scene occupancy.

Consider a single ray $\mathbf{\Gamma}(t) = \mathbf{r}_o + \hat{\mathbf{r}}t$ with $t \in [0, \infty)$. We sample the ray at a finite set of points $\{\mathbf{\Gamma}_i\}_{i=1}^N = \{\mathbf{\Gamma}(t_i)\}_{i=1}^N$ and query our volumetric belief state of the scene occupancy probability $p(\mathbf{\Gamma}_i) \in [0, 1]$. The ray will continue until it hits something, but since we are working with belief states, we need to quantify the probability of all possibilities. Let $p(\mathbf{\Gamma}_{1:i})$ denote the probability that the ray will pass through the volume at points $\{\mathbf{\Gamma}_j\}_{j=1}^{i-1}$ and then strike a surface at point $\mathbf{\Gamma}_i$ and stop. Treating the probability of occupancy at each point as independent, this can be computed as follows:

$$p(\mathbf{\Gamma}_{1:i}) = \prod_{j=1}^{i-1} \Big[1 - \ p(\mathbf{\Gamma}_j)\Big] p(\mathbf{\Gamma}_i) \tag{7}$$

This gives us the probability of N+1 different ray realizations, corresponding to the ray stopping at each point i, as well as the event where the ray completely passes through everything by setting $p(\mathbf{\Gamma}_{1:N+1}) = 1 - \sum_{i=1}^{N} \mathbf{\Gamma}_{1:i}$. For each possible ray realization, we accumulate the total uncertainty of the volume the ray passes through, given by the expression for binary Shannon entropy $\mathcal{H}(p(\mathbf{\Gamma}_i))$.

$$\begin{aligned} \mathcal{H}(p(\mathbf{\Gamma}_i)) = &- p(\mathbf{\Gamma}_i) \log p(\mathbf{\Gamma}_i) \\ &- (1 - p(\mathbf{\Gamma}_i)) \log(1 - p(\mathbf{\Gamma}_i)) \end{aligned} \tag{8}$$

Taking the expectation over all possible rays realizations for a total of K regularly sampled rays $\{\mathbf{\Gamma}^k(t)\}_{k=1}^K$, this gives us the following objective for active vision (see (7) and (8)):

$$f(\mathbf{r}) = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N+1} p(\mathbf{\Gamma}_{1:i}^k) \sum_{j=1}^{i} \mathcal{H}(p(\mathbf{\Gamma}_i^k)) \tag{9}$$

We use a total of $K = 23 \times 23 = 529$ rays sampled in a centered square grid with a vertical and horizontal FOV of 30 degrees. The rays are clipped to the interior of the object *ROI* and the box crop filter used during real point cloud acquisition (Section II-A).

D. Global and Hybrid Search Baselines

Owing to its gradient-based nature, our proposed control law from Section II-B is a local optimization algorithm for the objective function f that might get stuck in a local maxima or otherwise miss an informative camera view. To provide a comparison, we implement two baseline controllers based on conventional global viewpoint sampling [7], [8], [12]. These baseline controllers uniformly sample 5000 candidate poses $\mathbf{r}'$ from the full search space, selects the best one as a navigation goal $\mathbf{r}^*$, and sets the desired velocity twist of the secondary task $\mathbf{g}$ to the direction of the goal.

$$\mathbf{r}^* = \arg\max_{\mathbf{r}'} f(\mathbf{r}') \tag{10}$$

Note that this requires knowledge of the full search space geometry, including the ability to sample and reason about directions to a goal. For the table-top scenario considered here (Fig. 1), the search space is a partial sphere given by the center of the object and configured camera distance, which makes this possible by setting:

$$\mathbf{g} = \mathbf{g}^{global} = \alpha(\mathbf{r}, \mathbf{r}^*) \left(x^{global}, y^{global}, 0, 0, 0, 0\right) \tag{11}$$

Here, (x^{global}, y^{global}) is the camera-centric direction of the shortest path to the origin of the goal pose $\mathbf{r}^*$ along the tangent plane of hemispherical search space. Since the current camera frame $\mathbf{r}$ is already oriented with the z-axis pointing towards the center of the object, computing this direction amounts to mapping the goal origin $\mathbf{r}_\mathbf{o}^*$ into the coordinate frame of the current camera pose $\mathbf{r}$, extracting the x and y components and normalizing them to unit length. This makes $\mathbf{g}^{global}$ point in the direction of the shortest valid path to the goal. We control the magnitude of $\mathbf{g}^{global}$ with $\alpha(\mathbf{r}, \mathbf{r}^*)$, so that it is unit length when the distance between $\mathbf{r}$ and $\mathbf{r}^*$ is large and with linear decay to zero to slow down as $\mathbf{r}_o$ approaches $\mathbf{r}_o^*$ (10 cm threshold).

This allows us to create two new baseline controllers that we compare to our purely gradient-based **Local** controller from Section II-B2. **Global** search is a controller that exclusively uses the scheme described above. We sample a global goal pose, move directly to that goal, sample a new global goal, and repeat. **Hybrid** search combines the two approaches; first using local search until convergence, then switching to global search to get out of a potential local maxima before switching back to local search again and repeating. Both search strategies are implemented the exact same way as the gradient-based search, the only factor we vary is how to compute $\mathbf{g}$ allowing us to examine whether a local servo law can replace a global search.

Lastly, we define a termination criterion for the global and hybrid searches. A straightforward approach would be to stop when the objective value f no longer improves, but we found monitoring entropy around the object to be more robust. Specifically, we compute the average binary Shannon entropy across the set of voxels V_{ROI} inside the bounding box that surrounds the segmented object (Section II-A):

$$\mathcal{H}_{ROI} = \frac{1}{|\mathcal{V}_{ROI}|} \sum_{v \in \mathcal{V}_{ROI}} \mathcal{H}(p(v)) \tag{12}$$

For a fixed object, it would be reasonable to simply select a lower threshold and terminate the scan once $\mathcal{H}_{ROI}$ goes below this value. However, different objects produce different terminal entropy. The entropy is also not guaranteed to monotonically decrease, as the region of interest might change as more of the object is discovered, and noise might cause the occupancy probability of voxels to oscillate. Consequently, we instead examine whether the entropy curve has flattened out by evaluating $\mathcal{H}_{ROI}$ for the 10 most recent voxel grid updates and terminate if the standard deviation $\mathrm{stddev}\left(\{\mathcal{H}_{ROI}^i\}_{i=T-M}^{T}\right) < 0.001$.

III. Results and Discussion

A. Experimental Setup

Our physical setup consists of a 7-DOF Franka Emika Panda robotic arm equipped with a wrist-mounted Intel RealSense SR300 RGB-D camera. All experiments were conducted on a 40-core Intel Xeon E5-2630 CPU (2.2 GHz) workstation with 32 GB RAM, running Ubuntu 20.04.6 LTS with a real-time kernel ($\mathrm{v}5.15.111 - rt63$). The control laws are implemented with the ViSP library [29] (v3.6.0).

Due to kinematic constraints, the robot cannot cover a full hemisphere without reaching joint limits or colliding. Consequently, virtual barrier planes are defined relative to the base, with $\mathbf{g}$ projected back into the feasible set if violated (Fig. 3). For global search, goal candidates $\mathbf{r}'$ outside the workspace are rejected. Visual processing and control run asynchronously in separate threads for smoother control. Point cloud capture/processing runs at 0.7–1.3 Hz, while the servo law executes at 30–80 Hz (Local) and 40–125 Hz (Global), with goal sampling taking 1–3 s (Global). Hybrid executes at the rate of its current mode (Local/Global).

B. Evaluation Procedure

We scan eight test objects (Fig. 5) with the proposed Local strategy and the two baseline strategies Global and Hybrid. We perform three scans of the same object with each strategy, resulting in a total of 72 trials. The objects have a fixed pose and the camera pose is always initialized to approximately the same sideways view of the object. The raw, full-resolution RGB-D images are flushed to disk during the scans for offline postprocessing. The postprocessing procedure is similar to the online procedure described in Section II-A, including filtering, registration, accumulation with concatenation, voxel-based downsampling, and segmentation, but is implemented in python with algorithms from Open3D [30]. Additionally, we use a higher point cloud resolution of 1 mm and register all the captured point clouds simultaneously with multiway registration and pose graph optimization [31]. We then compare the high-resolution accumulated point clouds to ground-truth meshes of the objects. The comparison is done by first registering the mesh to the captured point clouds and then quantifying the discrepancy between the two.

The quantitative evaluation are based on Chamfer discrepancy [32], Hausdorf distance, and ground-truth coverage percentage.

$$\begin{aligned}
\text{Chamfer}(A, B) &= \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} d(a, b) \\
\text{Hausdorf}(A, B) &= \max_{a \in A} \min_{b \in B} d(a, b) \\
\text{Coverage}(A, B, l) &= \frac{100}{|A|} \sum_{a \in A} \mathcal{I}(\min_{b \in B} d(a, b) \leq l)
\end{aligned} \tag{13}$$

In all expressions, A corresponds to the set of points in the ground-truth mesh and B is the accumulated captured point clouds. The Chamfer and Hausdorf distance computes the respective average and maximum distance from each ground-truth point to its closest captured point. Both are one-sided variants, i.e., aggregated only over the (noise-free) ground truth points, as this lets us asses the completeness of the scan without distortions from outliers in the noisy captured cloud. Similarly, the Coverage metric computes the percentage of points in the ground truth mesh that are within a distance of ($r = 2.5$ mm) of a captured point. We approximate the

ground truth meshes with point clouds by sampling 10000 points from their surfaces uniformly. The full meshes are watertight and contain surfaces that are impossible to see, either because they are in contact with the table or otherwise have an orientation that requires a kinematically infeasible viewing angle. To prevent distances with points on these surfaces from polluting the measures, we ignore all mesh faces that have a normal vector of less than 45 degrees angle with the downward direction, i.e., retaining all surfaces that points mostly sidewards or upwards.

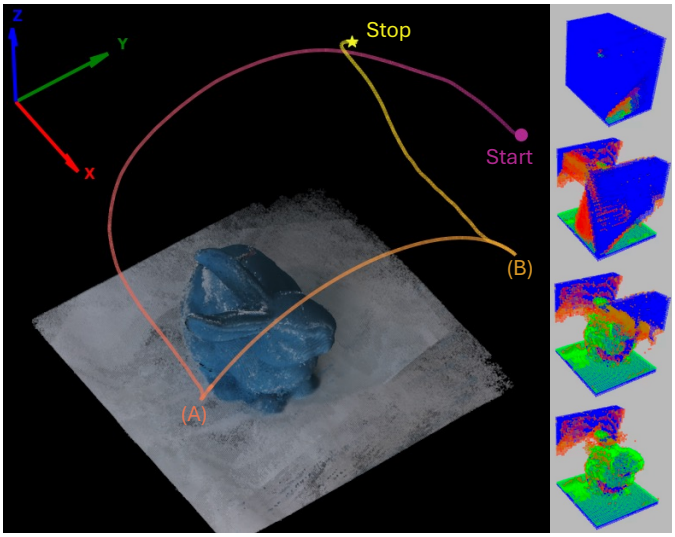

Fig. 3. **Left:** Evolution of the camera trajectory while scanning the Bunny object with the Local NBV-strategy. The coordinate frame corresponds to the base link of the robot. The two sharp turns at points marked (A) and (B) happen when the camera reaches the maximum permitted distance along the x-axis. The clean arc connecting the two points is caused by the controller sliding along the constraint plane with a projected gradient direction. **Right:** Occupancy probabilities around the object at four points during the scan. Green indicates high-confidence of occupancy, red and orange is medium confidence, and blue denotes unknown areas.

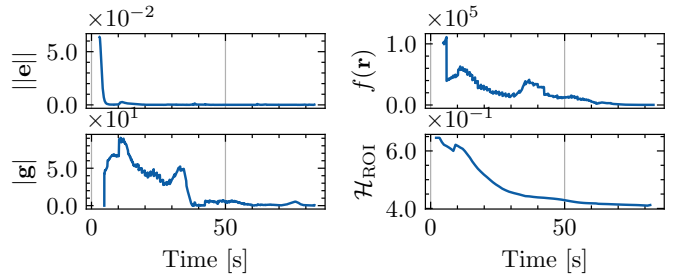

Fig. 4. Evolution over time of the task 1 error norm $||\mathbf{e}||$ (Eq. 2), the NBV objective value of the current camera pose $f(\mathbf{r})$ (Eq. 9), the resulting gradient norm of the secondary task $\mathbf{g}$ (Eq. 4, prior to clipping to unit norm), and the average voxel entropy in the region of interest $\mathcal{H}_{ROI}$ (Eq. 12). Values correspond to the same trial with the Bunny object and Local search as depicted in Fig. 3.

C. Real-world Evaluation Results

Fig. 3 visualizes one of the exploration trials of the complex Bunny object with the Local strategy for active vision. As the active vision proceeds, the uncertainty of the voxel grid is reduced, resulting in a nearly complete exploration of the object's 3D geometry at termination. The proposed method is seen to result in smooth camera trajectories (Fig. 3). We did occasionally observe oscillations in the camera movement, primarily as a result of noisy depth readings which caused the occupancy probabilities in the voxel map to fluctuate. These oscillations were more likely to occur towards the end of exploration, when most of the uncertainty that would otherwise dominate the gradient direction was exhausted.

Fig. 4 plots salient quantities for the trial visualized in Fig. 3. The top left plot shows the norm of the primary task error $||\mathbf{e}||$, which is shown to quickly converge (typically 1-2 seconds) and remains low and stable, even as the secondary task is activated. The top right plot shows the secondary task objective f evaluated at the current camera pose $\mathbf{r}$. It starts out high and decreases over time, with local peaks that occur whenever a new region of high uncertainty is revealed. The bottom left plot shows the norm of the gradient $\mathbf{g}$ used for the secondary task. The sharp drop in $||\mathbf{g}||$ around 38 seconds happens when the camera reaches the constraint plane for maximum x-position (distance forward from the robot base). From here, g is projected towards the feasible set, which substantially reduces its norm. Lastly, the bottom right plot shows the average entropy $\mathcal{H}_{ROI}$ in the voxels of the occupancy grid that surrounds the object. Even though this quantity is not used to terminate the Local strategy shown here, we see that it gently decreases and its flattening is a reliable indicator of a complete exploration trajectory.

Fig. 5 shows the physical objects used (top row), along with accumulated point clouds resulting from exploration trials with the Local strategy. The bottom row plots Chamfer distance as a function of time for each trial and object. The Local strategy terminates the fastest compared to other strategies with an average runtime of approximately 42 s, while Global and Hybrid use on average 57 s and 67 s respectively. Similarly, the average trajectory lengths are 0.8 m, 0.8 m, and 1.2 m for Local, Global, and Hybrid.

Table I summarizes the performance metrics (Eq. 13) computed over the final point clouds for each object and search strategy. The average Chamfer distance is 1.6 mm for the Hybrid strategy, which is slightly better than for the Local and Global strategy (both 1.8 mm on average), justifying the longer exploration time and distance. However, looking at the line plots at the bottom of Fig. 5, it is clear that the majority of decrease in Chamfer distance happens early on. The plots show a consistent profile across all objects: the Chamfer distances start at around $10 - 25$ mm and progressively decrease to around $0.5 - 4$ mm upon convergence, depending on object. This convergence is typically reached within 50 seconds of exploration time. While the Hybrid strategy appears to obtain slightly better final performance on some of its longer runs, particularly on the bunny, the improvements are marginal. This aligns with our qualitative assessment — all strategies initially move with purpose towards high-uncertainty regions, but only make small adjustments at the end of the exploration. The Local and Hybrid strategies typically result in faster exploration of the object, evidenced by a faster decrease in Chamfer distance. This is mainly because the Global strategy

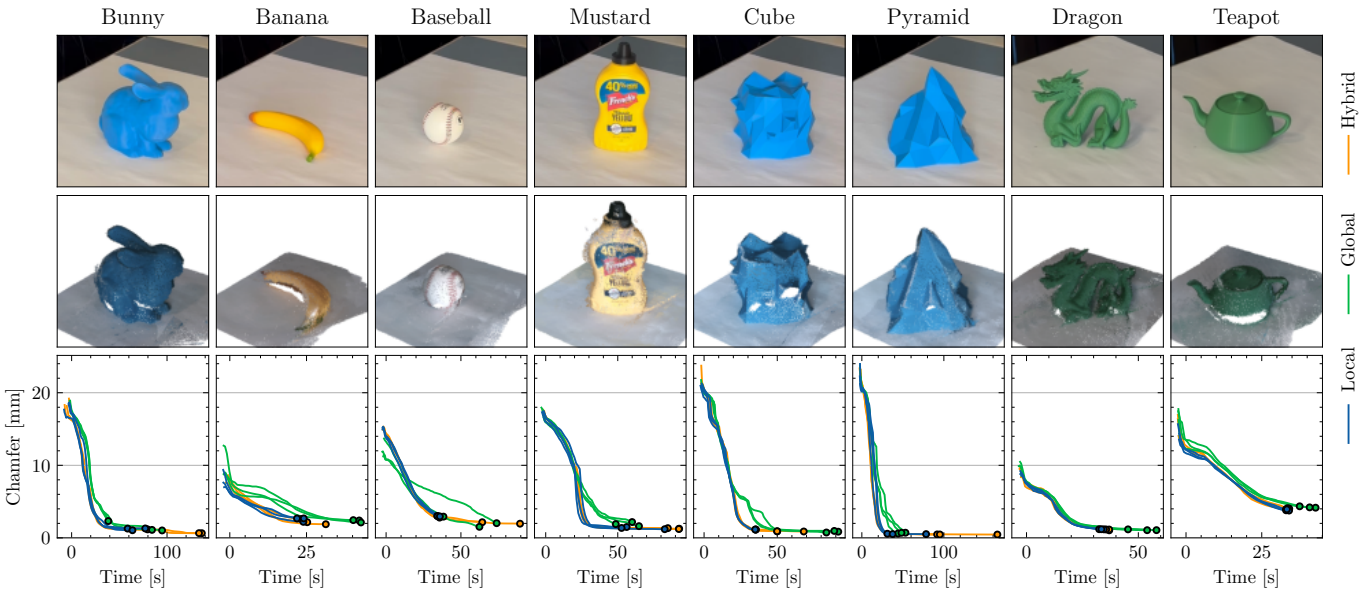

Fig. 5. **Top**: Images of the evaluation objects, including three everyday YCB objects [33] and five 3D-printed objects with complex geometry, three test objects from [9]. **Middle**: Resulting cleaned and accumulated point clouds from a scan with the Local NBV-strategy. **Bottom**: Chamfer distance as a function of scan time for all trials, color-coded by the NBV-strategy. Dots indicate termination of a trial.

TABLE I
QUANTITATIVE RESULTS FOR EACH OBJECT AND CONTROL STRATEGY. EACH NUMBER IS THE AVERAGE OF THREE RUNS (± STANDARD ERROR, N=3).

Metric	Search Type	Bunny	Banana	Baseball	Mustard	Cube	Pyramid	Dragon	Teapot
Chamfer [mm]	Global	1.5 ± 0.4	2.3 ± 0.1	$\mathbf{2.2 \pm 0.4}$	1.9 ± 0.2	$\mathbf{0.9 \pm 0.1}$	0.7 ± 0.0	$\mathbf{1.1 \pm 0.0}$	4.2 ± 0.1
	Hybrid	$\mathbf{0.8 \pm 0.2}$	$\mathbf{2.2 \pm 0.2}$	2.3 ± 0.3	$\mathbf{1.3 \pm 0.0}$	0.9 ± 0.0	$\mathbf{0.5 \pm 0.0}$	1.1 ± 0.0	4.0 ± 0.0
	Local	1.2 ± 0.1	2.5 ± 0.2	3.0 ± 0.0	1.4 ± 0.1	1.1 ± 0.0	0.5 ± 0.0	1.2 ± 0.0	$\mathbf{3.8 \pm 0.1}$
Hausdorf [mm]	Global	23.2 ± 1.5	$\mathbf{14.2 \pm 0.4}$	$\mathbf{19.8 \pm 4.6}$	17.4 ± 2.4	15.2 ± 1.5	15.4 ± 1.5	$\mathbf{17.3 \pm 0.3}$	$\mathbf{39.3 \pm 0.2}$
	Hybrid	$\mathbf{12.9 \pm 2.2}$	14.5 ± 0.4	22.9 ± 0.6	$\mathbf{14.2 \pm 0.4}$	$\mathbf{14.1 \pm 0.9}$	$\mathbf{6.7 \pm 0.5}$	19.0 ± 0.6	39.4 ± 0.7
	Local	17.8 ± 0.4	15.9 ± 0.4	24.8 ± 0.9	14.9 ± 1.3	16.1 ± 0.6	8.4 ± 0.6	18.8 ± 0.8	40.1 ± 0.7
Coverage [%]	Global	88.2 ± 4.4	69.4 ± 1.6	77.3 ± 3.1	80.2 ± 2.6	94.9 ± 0.9	97.4 ± 0.5	$\mathbf{90.8 \pm 0.4}$	67.4 ± 0.6
	Hybrid	$\mathbf{95.1 \pm 2.7}$	$\mathbf{75.4 \pm 2.9}$	$\mathbf{78.8 \pm 3.1}$	90.6 ± 1.3	$\mathbf{95.0 \pm 0.3}$	$\mathbf{99.7 \pm 0.0}$	90.6 ± 0.6	67.5 ± 0.4
	Local	90.1 ± 1.1	71.2 ± 1.6	72.0 ± 0.2	$\mathbf{90.8 \pm 1.1}$	92.2 ± 0.2	99.1 ± 0.2	89.6 ± 0.4	$\mathbf{68.9 \pm 0.6}$

frequently has to stop to sample new goals.

Perhaps the most striking observation is that the Local strategy rarely gets stuck and remains competitive, despite only steering with local gradient information. In general, we observe that it would produce consistent trajectories that covered the object from the same sides as the Global and Hybrid strategies. One exception is the Cube object, where the Local strategy would move in a straight arc scanning the object from the right, top, and left before terminating. In contrast, the Hybrid strategy would first follow the same path, but then sample a goal that allowed it to also view the object from its front side. Nevertheless, these results demonstrate that gradient-based NBV is surprisingly resistant to Local maxima for the considered scenario. We conjecture that this is because our information-oriented NBV objective function f is non-stationary over time in a way that makes it well-suited for gradient-based search. That is, even though the search greedily maximizes f, owing to the closed-loop nature of our approach, new information constantly arrives and, if a peak is reached, eventually turns it into a valley, allowing the robot to keep moving (see e.g. sharp turns in Fig. 3). Our real-world findings are also consistent with trends in the simulation experiments from [21], which show that although global sampling strategies for NBV may have a slight asymptotic advantage, gradient-based local search is a viable alternative for efficient exploration.

D. Challenges and Future Work

One of the key challenges encountered in our real-world experiments relates to the workspace constraints and kinematic reachability of the Franka Emika Panda robot arm. In particular, limited reach, self-collision, and collision with the table made certain camera views unattainable. Additionally, excessive exploration time would sometimes lead to bad joint configurations. In principle, the control law could be augmented with extra terms for mitigating this issue. The primary and secondary tasks only places a 5-DoF constraint on camera movement, so the flexibility afforded by rotation around the camera viewing axis could be exploited for better joint limit avoidance. For future work, it would be interesting to deploy the controller on robotic platforms with fewer kinematic constraints, for example autonomous underwater vehicles. Since

the servo law generates velocities in the camera frame, such a change could be implemented with minimal changes to the core control logic presented here. Additionally, we intend to integrate the method into a complete and autonomous pipeline for 3D asset generation.

IV. Conclusion

In this letter, we presented a closed-loop, real-time active vision framework for 3D reconstruction of unknown objects using visual servoing for continuous NBV planning. Our method is agnostic to the exact form of the NBV objective function and couples a primary visual servoing task, that keeps the camera focused on the object, with a secondary task that generates exploratory motion by ascending the NBV gradient. Extensive real-world experiments with a physical robot and a diverse set of objects show that our framework generates smooth exploration trajectories. We show that our local gradient-based NBV-strategy, without any prior knowledge, produces exploration trajectories with strong object coverage that are highly competitive with global sampling strategies that leverage prior knowledge about the search space. Unlike prior studies conducted solely in simulation, we demonstrate an active vision framework on a physical robot-system with a wide range of objects of varied shape, size and geometry. The proposed framework is applicable beyond generic robotic manipulation, with potential use in manufacturing, robotic food inspection and manipulation, and cultural heritage inspection, where it can replace preprogrammed scanning systems by autonomously adapting viewpoint trajectories. To the best of our knowledge, no existing work achieves closed-loop active perception that simultaneously integrates a continuous NBV-planning and 3D reconstruction in real-time on a real-world robotic system.

References

[1] U. J. Isachsen, T. Theoharis, and E. Misimi, "Fast and accurate GPU-accelerated, high-resolution 3D registration for the robotic 3D reconstruction of compliant food objects," *Computers and Electronics in Agriculture*, vol. 180, pp. 1–8, 2021.

[2] R. Zeng, Y. Wen, W. Zhao, and Y.-J. Liu, "View planning in robot active vision: A survey of systems, algorithms, and applications," *Computational Visual Media*, vol. 6, no. 3, pp. 225–245, 2020.

[3] Y. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *Int. Journal of Computer Vision*, vol. 1, pp. 333–356, 1988.

[4] E. Marchand and F. Chaumette, "Active vision for complete scene reconstruction and exploration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 1, pp. 65–72, 1999.

[5] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The Int. Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.

[6] W. R. Scott, G. Roth, and J. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys*, vol. 35, no. 1, pp. 64–96, 2003.

[7] C. L. Connolly, "The determination of next best views," in *IEEE Int. Conf. on Robotics and Automation (ICRA'85)*, vol. 2, 1985, pp. 432–435.

[8] L. Hou, X. Chen, K. Lan, R. Rasmussen, and J. Roberts, "Volumetric next best view by 3D occupancy mapping using Markov chain Gibbs sampler for precise manufacturing," *IEEE Access*, vol. 7, pp. 121 949–121 960, 2019.

[9] M. Mendoza, J. I. Vasquez-Gomez, H. Taud, L. E. Sucar, and C. Reta, "Supervised learning of the next-best-view for 3D object reconstruction," *Pattern Recognition Letters*, vol. 133, pp. 224–231, 2020.

[10] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3D object reconstruction," *Autonomous Robots*, vol. 42, no. 2, pp. 197–208, 2018.

[11] J. I. Vasquez-Gomez, L. E. Sucar, R. Murrieta-Cid, and E. Lopez-Damian, "Volumetric next-best-view planning for 3D object reconstruction with positioning error," *Int. Journal of Advanced Robotic Systems*, vol. 11, pp. 159:1–159:13, 2014.

[12] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3D reconstruction," in *IEEE Int. Conf. on Robotics and Automation (ICRA'16)*, 2016, pp. 3477–3484.

[13] J. Daudelin and M. Campbell, "An adaptable, probabilistic, next best view algorithm for reconstruction of unknown 3D objects," *IEEE Robotics and Automation Letters*, vol. 10, pp. 1–8, 2017.

[14] M. D. Kaba, M. G. Uzunbas, and S. Lim, "A reinforcement learning approach to the view planning problem," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 5094–5102.

[15] M. Breyer, L. Ott, R. Siegwart, and J. J. Chung, "Closed-loop next-best-view planning for target-driven grasping," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'22)*, 2022, pp. 1411–1416.

[16] D. Morrison, P. Corke, and J. Leitner, "Multi-view picking: Next-best-view reaching for improved grasping in clutter," in *Int. Conf. on Robotics and Automation (ICRA'19)*, 2019, pp. 8762–8768.

[17] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3D object models using next best view manipulation planning," in *IEEE Int. Conf. on Robotics and Automation (ICRA'11)*, 2011, pp. 5031–5037.

[18] S. A. Kay, S. Julier, and V. M. Pawar, "Semantically informed next best view planning for autonomous aerial 3D reconstruction," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'21)*, 2021, pp. 3125–3130.

[19] R. Menon, T. Zaenker, N. Dengler, and M. Bennewitz, "NBV-SC: Next best view planning based on shape completion for fruit mapping and reconstruction," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'23)*, 2023, pp. 12 054–12 061.

[20] H. Dhami, V. D. Sharma, and P. Tokekar, "MAP-NBV: Multi-agent prediction-guided next-best-view planning for active 3D object reconstruction," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'24)*, 2024, pp. 5724–5731.

[21] A. K. Burusa, E. J. van Henten, and G. Kootstra, "Gradient-based local next-best-view planning for improved perception of targeted plant nodes," in *IEEE Int. Conf. on Robotics and Automation (ICRA'24)*, 2024, pp. 15 854–15 860.

[22] S. H. Gazani, M. Tucsok, I. Mantegh, and H. Najjaran, "Bag of views: An appearance-based approach to next-best-view planning for 3D reconstruction," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 295–302, 2024.

[23] R. Monica and J. Aleotti, "Contour-based next-best view planning from point cloud segmentation of unknown objects," *Autonomous Robots*, vol. 42, pp. 443–458, 2018.

[24] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *IEEE Int. Conf. on Robotics and Automation (ICRA'11)*, 2011, pp. 1–4.

[25] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in 3^{rd} *IEEE Int. Conf. on 3D Digital Imaging and Modeling*, 2001, pp. 145–152.

[26] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[27] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

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

[29] E. 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, pp. 40–52, 2005.

[30] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv preprint arXiv:1801.09847*, 2018.

[31] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 5556–5565.

[32] A. Bakshi, P. Indyk, R. Jayaram, S. Silwal, and E. Waingarten, "Near-linear time algorithm for the Chamfer distance," *Advances in Neural Information Processing Systems*, vol. 36, pp. 66 833–66 844, 2023.

[33] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in Manipulation Research: The YCB Object and Model Set and Benchmarking Protocols," *IEEE Robotics and Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.