INGRID: Interactive Geometry and Instance Identification for Occluded Scenes

Junho Lee*, Sangmin Kim*, Yonghyeon Lee† and Young Min Kim*
*Seoul National University
†Massachusetts Institute of Technology

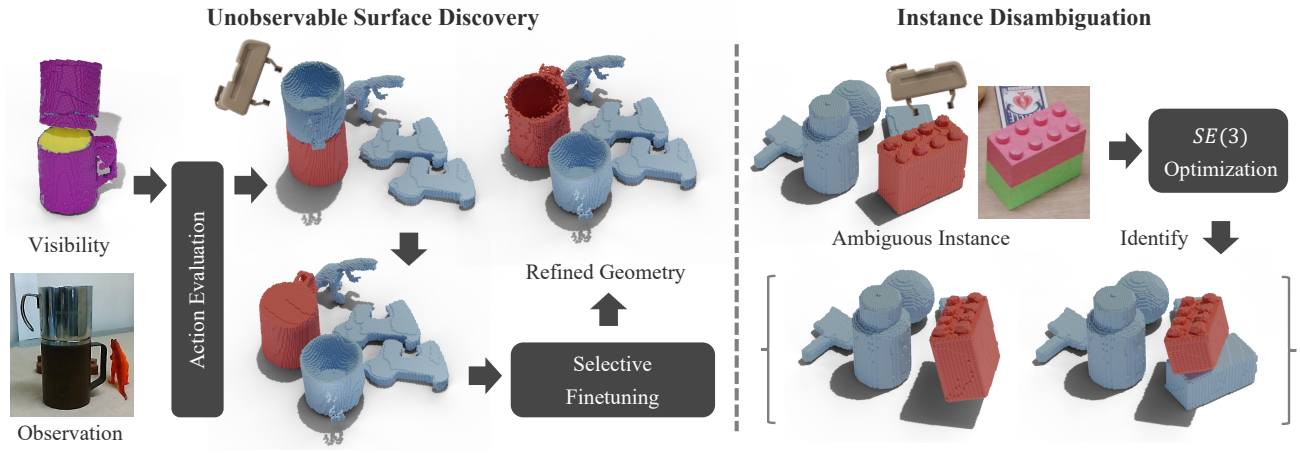


Fig. 1: Overview of INGRID (Interactive Geometry and Instance Identification). INGRID actively discovers previously unobservable surfaces by executing actions with a robot arm that maximize scene visibility. Through selective finetuning, it efficiently refines the underlying geometry. Furthermore, by interacting with the environment, INGRID resolves instance-level ambiguities.

***Abstract*—While identifying object instances and accurately reconstructing their geometry are fundamental to manipulation, *passive* visual observation in static scenes is inherently constrained by occlusions, which gives rise to both geometric and semantic ambiguities, particularly in complex multi-object settings. We present INGRID (INteractive GeometRy and instance IDentification), a system in which a robotic manipulator *actively* interacts with the scene to reveal previously hidden surfaces and disambiguate object instances and geometries. The key challenge lies first in deciding where to interact in order to induce the most informative scene changes. A subsequent challenge is then to efficiently update both the object instances and their geometries once new visual observations become available. To address these challenges, INGRID follows a four-step pipeline: (i) constructing an instance candidate tree, (ii) optimizing informative actions, (iii) identifying instances under rigid-body assumptions, and (iv) selectively finetuning geometry. Together, these steps enable efficient, interaction-driven geometry and instance identification. Our method proves robust and practical in complex multi-object scenes with severe occlusions, both in simulation and in the real world.**

I. Introduction

The ability to reconstruct object geometry and identify object instances from visual observations is a fundamental perception capability for robotic manipulation. An object instance here refers to a rigidly connected entity whose geometry and identity must be distinguished from other objects in the scene. Such instance-level understanding is crucial for many downstream tasks, including integrating object-centric grasping modules [40, 16, 30], rapidly detecting scene dynamics [10], and bridging the sim-to-real gap [36].

Given multi-view RGB images, recent works construct static 3D scene representations by leveraging Neural Radiance Fields (NeRFs) and their variants [38, 23, 27, 34]. These representations often include density fields, which capture the underlying object geometry and can be converted into surface point clouds – even in challenging scenarios such as transparent objects that conventional depth cameras cannot handle [34]. Furthermore, 2D segmentation masks [31] are integrated to estimate coherent 3D object instances in multi-object scenes [7, 29]. Taken together, these advances provide a powerful framework for simultaneous geometry reconstruction and object instance identification from multi-view RGB images.

However, in the presence of object occlusions, passive visual observation – i.e., relying solely on static images – inevitably leads to both geometric and instance ambiguities. For example, as shown in Fig. 1, when two cups are stacked vertically, the lower cup is occluded, making it ambiguous whether its interior is hollow or solid. Similarly, in the case of the block on the right, the object actually consists of two separable blocks, yet their similar semantic features make it difficult to determine whether they should be identified as one block

or two distinct instances. A natural and promising direction is to enable the robot to actively interact with the scene, thereby revealing hidden information and resolving such ambiguities.

To this end, we introduce INGRID (INteractive GeometRy and instance IDentification), illustrated in Fig. 1, an efficient framework that (i) enables a robot to autonomously induce scene changes through interaction and acquire new visual observations, (ii) identifies object instances, and (iii) selectively refines object geometry.

The first challenge is to identify the most ambiguous region of the 3D scene, where spatial interactions such as pushing or pick-and-place would yield the greatest information gain. To this end, we construct an instance candidate tree from three levels of SAM mask predictions [12], where each node is assigned a point cloud representing an instance candidate and child nodes correspond to subsets of their parents. The action space – candidate spatial perturbations – is then defined over the tree and restricted to its leaf nodes. Formulated as an optimal search problem, the selected action is the one that maximizes the newly revealed surface area.

Given a changed scene and new visual observations, the second challenge is to efficiently identify the correct instances from the tree and refine their geometry. Assuming rigid bodies, we identify instances by jointly optimizing their SE(3) transformations. Starting from top-level nodes – which may group multiple adjacent objects and cause large errors – we iteratively split into lower-level nodes until the optimization error falls below a threshold. The selected instance is then refined geometrically in a selective manner for efficiency.

Our extensive experiments demonstrate the effectiveness of our methods in both simulated and real-world scenarios, validating a framework capable of identifying object instances and geometry of unknown 3D objects in occlusion-prone scenes through interaction. Our contributions can be summarized as follows:

- An interaction algorithm to autonomously induce change for occluded objects;
- An efficient instance-wise geometric finetuning scheme utilizing a visibility metric;
- An optimization formulation to jointly optimize the instances and transform given a few images after change.

II. Related work

A. From NeRFs to Feature Fields for Robot Applications

Neural Radiance Fields (NeRFs) train a volumetric representation, namely the color c and density σ of the 3D scene, to synthesize novel view images of the scene [38]. While NeRF was originally developed for novel view synthesis, it has inspired a line of research that utilizes field representations for various vision-based robotic tasks. For example, the Signed Distance Field (SDF) can be seamlessly integrated into NeRF frameworks through a simple conversion formula [52, 49]. Consequently, in robotics applications, SDF has been further employed for grasping [48, 15] and SLAM tasks [25, 56]. Additionally, the trained density σ provides geometric estimates for navigation [1, 32] and grasping [22, 27, 28]. The volumetric aggregation of NeRF can also be extended to incorporate rich image features from vision foundation models. For instance, instead of RGB colors, one can use surface normals to build a coherent volume for grasping applications [34]. Moreover, several works apply rendering loss on feature images to lift CLIP features into a 3D volume [26, 47, 44], enabling interaction through language commands. Several works closely related to our paper leverage segmentation labels [31] to construct novel feature fields that capture instance information [12, 7, 29, 20, 53, 13, 55, 14, 51]. This approach results in a field that simultaneously represents both geometry and instance identity. Specifically, by employing a contrastive loss, these methods learn an affinity field [7] that maps two points from the same object to similar feature values. To further segment these fields into discrete clusters in 3D space, clustering algorithms such as HDBSCAN [11] are applied. While numerous studies have explored various field representations as discussed above, passive observation of static scenes still leads to ambiguities in both geometry and instance identification. To address this, introducing controlled scene changes and leveraging these changes to recognize and update the field accordingly remain largely underexplored.

B. Active Perception

Active perception [4, 5, 6] refers to a class of methodologies in which a robot leverages its own actions to improve its understanding of the environment. It is a well-established area of research with broad applications [9], including pose estimation [50], object segmentation [45], articulation [41] and dynamics modeling [2], and the learning of manipulation [33] or grasping strategies [42]. Recent works have extended active perception toward high-level scene understanding by incorporating large language models and structured heuristics. These systems can perform complex interactions to discover objects and construct symbolic representations such as scene graphs [24]. However, these approaches often rely heavily on precise prompt engineering and assume access to near-perfect segmentation modules—conditions that may not hold in real-world settings. Among related approaches, the work most aligned with our philosophy is UncOS [18], which addresses instance ambiguity in cluttered scenes by using robotic pushing. While UncOS addresses a similar problem, its setting is limited to single-view images, which hampers accurate detection of instance-wise object geometry under occlusion. In contrast, INGRID leverages multi-view observations to aggregate instance and geometric information, enhancing its robustness to occlusion.

III. Initial Field Representation

In this section, we employ *normal*, *density*, and *feature field* representations to obtain initial estimates of object instance and geometry. Building on prior work [34], we use a voxelized surface normal field $n(x)$ and a volume density field $\sigma(x)$ for $x \in \mathbb{R}^3$ to capture object geometry[1]. In this work, inspired by

[1]The density field $\sigma(x)$ assigns zero values to regions outside the surface, while the normal field $n(x)$ is defined only on object surface points.

Contrastive Lift [7], we further incorporate 3 levels (coarse, mid, fine) of feature fields $F_c(x), F_m(x), F_f(x)$ defined on surface points to encode instance-related information. By clustering in this feature vector space, we can then identify object instances.

While our main contribution is an interactive method to obtain the geometry and instance of objects in occlusion-prone scenes, in this section, we first describe how to *initialize* geometry and instance from multi-view, relatively dense RGB images, following previous works [34, 35, 7]. This section consists of the following three subsections: (i) input preprocessing, (ii) normal and density fitting loss, and (iii) feature learning loss.

A. Input Preprocessing

Given raw multi-view RGB images with camera poses, we preprocess these images to generate mask and surface normal images using pre-trained models. Specifically, we employ SAM [31] to map each RGB image I to three sets of masks with different granularities: $M_c(I)$, $M_m(I)$, and $M_f(I)$, representing coarse, mid, and fine-level masks, respectively. In this process, we use SAM's point query method, similar to the preprocessing approach in LangSplat [43]. We then use DSINE [3] to estimate surface normals denoted by $N(I)$.

B. Normal and Density Fitting Loss

We utilize the masks $M_c(I)$, $M_m(I)$, and $M_f(I)$ and the surface normals $N(I)$ to train the normal and density fields, $n(x)$ and $\sigma(x)$, which represent the geometry of the objects, depicted in Fig. 2. The two fields are approximated using voxel grid representations and fitted to the predicted normal and segmentation masks, following prior work [34]. While we adopt two key loss terms from the prior work [34] – (i) the normal vector rendering loss and (ii) the density penalization loss for background pixel rays – in this section, we introduce *density-normal consistency regularization loss* to further enhance geometric accuracy.

We empirically observe that for highly concave objects, the normal rendering loss alone tends to cause $n(x)$ to overfit the rendered normal images while significantly violating geometric consistency with $\sigma(x)$, leading to distorted results. To address this issue, we design a density-normal consistency regularization loss for each ray r, inspired from MonoSDF [54]. We define the loss as the difference between the normal vector obtained from the gradient of the density field σ and the predicted normal N. Denoting the accumulated transmittance along the ray r as

$$T(r,t) = \exp(-\int_{t_n}^{t} \sigma(r(s))ds), \tag{1}$$

the loss can be expressed:

$$L_{\text{normal}}(r) := \left\| \frac{\int_{t_n}^{t_f} T(r,t)\nabla\sigma(r(t))\sigma(r(t))dt}{\left\|\int_{t_n}^{t_f} T(r,t)\nabla\sigma(r(t))\sigma(r(t))dt\right\|} - N(r) \right\|. \tag{2}$$

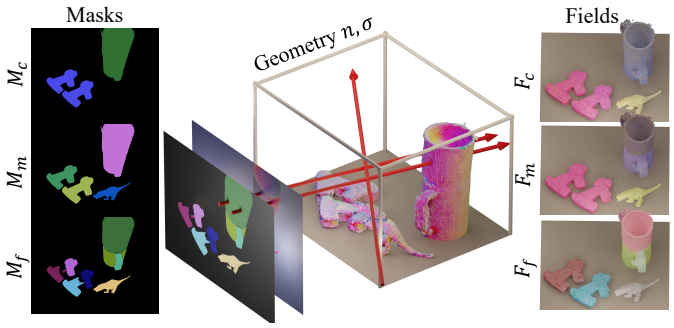


Fig. 2: Process to learn field. From coarse (M_c), mid (M_m), fine (M_f) granularity masks and normal images, we learn three instance feature fields (F_c, F_m, F_f) each reflecting the granularity on shared geometry (n, σ).

C. Feature Learning Loss

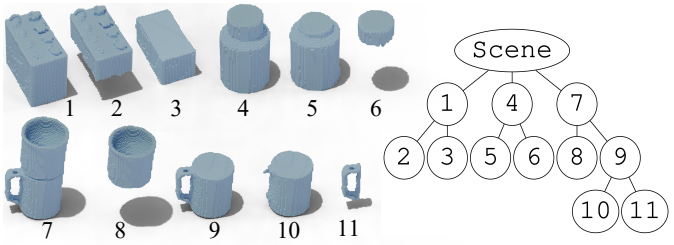
Given the density field $\sigma(x)$, we use the segmentation masks of three different levels $M_c(I), M_m(I), M_f(I)$ to train three hierarchical affinity fields $F_c(x), F_m(x), F_f(x)$ as in Fig. 2. The three different hierarchies of fields are trained simultaneously but independently, applying a contrastive loss formulation [7] to the corresponding granularity of the mask. In brief, contrastive loss encourages rays of pixels with the same mask (positive pairs) to have similar features, while ensuring the rays from different mask labels (negative pairs) have distant feature vectors. At each training step, we sample 256 pairs of rays from a single viewpoint to construct positive and negative pairs for the triplet loss [46].

IV. Interactive Geometry and Instance Identification

In this section, we introduce INGRID (Interactive Geometry and Instance Identification), a framework for identifying both the geometry and instance of objects in occlusion-prone scenes through interaction. Building upon the field representations (F_c, F_m, F_f) described in Section III, we first generate a hierarchy of instance candidates in a tree-like structure (Section IV-A). Based on the tree, we then define an optimization-based interaction algorithm to identify informative interactions that reveal previously unobservable surfaces (Section IV-B). After executing the chosen action, we simultaneously identify each instance and estimate its corresponding $SE(3)$ transformation given sparse RGB images (Section IV-C). Finally, we refine the geometry of the newly visible regions for efficiency, a process we refer to as selective finetuning (Section IV-D) to obtain instance-wise geometry.

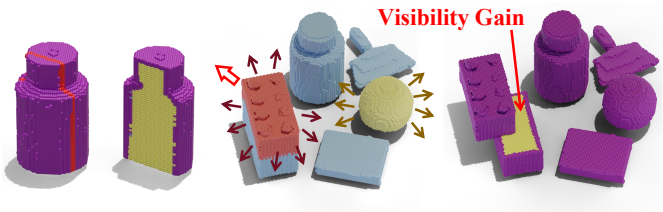
A. Instance Candidate Tree Construction

After obtaining the geometric layout of the scene (σ, n), the additional information on instance clusters can provide structure, such that we can design the action sequences and perform desired tasks. While the instance geometry and boundaries can be ambiguous under static passive observation, the hierarchical fields (F_c, F_m, F_f) aggregate the distribution of potential instances from multiple viewpoints. We further cluster them into



(a) Geometry Per Node (b) Instance Candidate Tree

Fig. 3: Instance candidate tree and its nodes. The Instance candidate tree (b) contains geometry of the candidates in its nodes (a), with the child node included in its parent node.



(a) Visibility U (b) Action Space A (c) Updated Visibility $\widetilde{U_a}$

Fig. 4: Interaction process visualization. We compute the visibility U encoding how much each voxel was observed during training for each object voxels (a). For each action a in the action space A (b), we obtain the updated visibility $\tilde{U}_a$ (c) and evaluate the action value proportional to the visibility gain (highlighted in yellow).

a tree structure with discrete granularity, forming the *instance candidate tree* as follows. First, we extract the point cloud of the objects by thresholding σ and perform clustering with HDBSCAN [37] on each F_c, F_m, F_f. This gives us multiple candidate clusters of varying sizes from different hierarchies of fields with possible overlap. Second, we sort all clusters (irrespective of F_c, F_m, F_f) in descending order based on the sizes of the bounding boxes of their respective point clouds. We then apply an iterative tree generation algorithm. Starting from the largest cluster, the process sequentially determines whether each cluster should be (1) added as a child of an existing node, (2) ignored since it is a duplicate of an existing node, or (3) assigned as a new root node, thereby expanding the tree accordingly[2]. As illustrated in Fig. 3, each node in the tree represents an instance candidate, with child nodes corresponding to subsets of their respective parent nodes.

B. Algorithms for Determining Where-to-Interact

This section proposes an algorithm that interacts with objects in the scene to reveal unobserved or occluded surfaces or discern object instances. Assuming an instance geometry rigidly moves together under interaction, we can define a set of actions from the instance tree structure of Section IV-A. From the set of action spaces, INGRID chooses the action to maximize observation of unseen geometry.

First, in order to track unobservable parts, we propose a 3D visibility field that represents how much each point of the objects was observable during the field training process of Section III. Given the standard definition of transmittance in Eq. (1), we define the visibility U (depicted in Fig. 4 (a)) for a voxel (i, j, k) as:

$$U(i,j,k) = \max_{r \in R_{ijk}} \big(T(r,t^*)\sigma(r(t^*))\big), \tag{3}$$

where R_{ijk} is the set of training rays that pass through the voxel (i, j, k). t^* is defined for each ray $r(t) = o + td$ with origin o and direction d such that $r(t^*) = (i, j, k)$. U is defined on objects voxels only, obtained by thresholding on σ, and is calculated for all object voxels once, after training the geometry n, σ. Voxels with lower U indicate less visibility (i.e., unobserved during training), and should therefore be prioritized for uncovering through interaction.

The action space A – a set of candidate spatial perturbations – is defined based on the instance candidate tree. An action $a \in A$ consists of a target instance node selected from the set of leaf nodes, along with a displacement vector on the x–y plane, with z axis pointing up. The action space A spans all leaf nodes and 12 predefined directions with 20 distance magnitudes ranging from 1 cm to 40 cm (Fig. 4 (b)). For each $a \in A$, the visibility field U is updated along with $\sigma(x)$ assuming a rigid body transform, which we denote by $\tilde{U}_a$. After displacement, the node's new region inherits original visibility values, and its former region is set to 0 ($\sigma = 0$) as it becomes empty (Fig. 4 (c)).

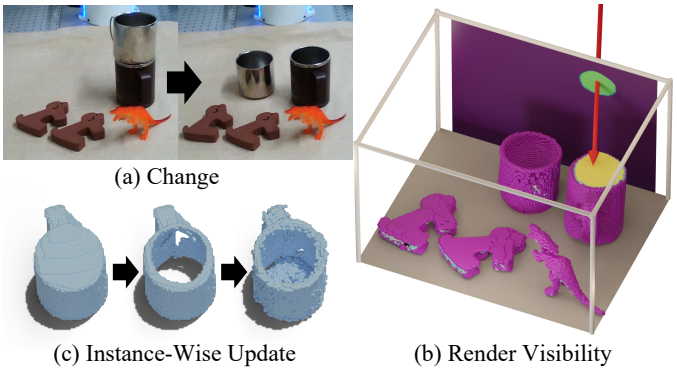
We define a value function $Q(a)$ that quantifies the amount of newly visible surface area revealed by an action a. Given a set of camera viewpoints V, the value function is defined as:

$$Q(a) := \sum_{v \in V} \mathrm{proj}_v(\tilde{U}_a) + C(a), \tag{4}$$

where proj_v denotes the projected visible surface area from viewpoint v (Fig. 5 (b)), and $C(a)$ is a penalty term set to $-\infty$ for actions that result in collision. We used cameras at an altitude of 45 degrees, equally spaced around the objects. Within the action space A defined above, we select the action that maximizes the value function, i.e., $\arg\max_{a \in A} Q(a)$. For scenarios with multiple instances, we select one action per connected component in the instance candidate tree.

To execute the selected action on a physical robot, we design a pick-and-place pipeline using a Franka Panda robot arm. Grasp poses for the target node are generated via AnyGrasp [17], and the candidate with the highest predicted score is selected. Joint position and velocity trajectories are then computed using PyBullet Planning [19] to guide the robot to the grasp pose while ensuring collision avoidance. These trajectories are executed on the physical robot using a joint-space PD controller. Following the grasp, the gripper is raised 20 cm, the 2D action $\arg\max_{a \in A} Q(a)$ is applied, the gripper descends 20 cm to its original height, and the object is released.

[2]We utilize Intersection over Union (IoU) between the existing cluster and the one being added.



(a) Change

(c) Instance-Wise Update

(b) Render Visibility

Fig. 5: Selective geometric finetuning process. After a scene change (a), we calculate the visibility of each surface (yellow in (b)) that represents newly visible parts. For instances with previously unobservable parts, we perform finetuning (c) to refine the geometry.

C. Instance Identification under Rigid-Body Assumption

Execution of the selected actions $\arg\max_{a \in A} Q(a)$ separates objects and uncovers new surfaces. We assume the individual instances move rigidly under interaction, and update the scene configuration accordingly. To do so, we need to identify the correct set of instances from the instance candidate tree and their corresponding $SE(3)$ transforms from image observations.

Let $v = 1, \ldots, V$ be the index for the new observations, each captured from a total of V pre-defined viewpoints in Section IV-B, and m_v be the 2D projection of the point cloud in the v-th view. Let X_i denote a 3D cluster or point cloud in the tree, where $i = 1, \ldots, N$, and N is the total number of nodes in the tree. Let T_i be the rotation and translation parameters for X_i, and when X_i is transformed by T_i and projected onto the image plane in the v-th direction, it forms a 2D point cloud, which we denote by $\text{proj}_v(T_i X_i)$. Also, we define the set of nodes selected to be instances as I.

Starting from the coarse granularity, we progressively adjust the node combination in I along with the rotation and translations $\{T_i\}$ to match the 2D observations:

$$\min_{\{T_i : i \in I\}} \sum_{v=1}^{V} \sum_{i \in I} d\big(\text{proj}_v(T_i X_i), m_v\big), \tag{5}$$

where $d(\cdot, \cdot)$ is the Chamfer distance metric between two 2D point clouds. For example, in the scene of Fig. 3, we first try matching the instance set of $I = \{1, 4, 7\}$. If the objective function is minimized below a predefined threshold, we stop and return the instances I and transforms $\{T_i\}$. Otherwise, we split one of the nodes and redefine I, e.g., $\{1, 4, 7\} \rightarrow \{2, 3, 4, 7\}$, then solve the optimization again. We repeat this process until the alignment error is sufficiently low.

D. Selective Geometric Finetuning

In this section, we introduce an efficient dynamic scene update scheme that leverages instance-wise transforms T_i and visibility information U derived from the few-shot input described in Section IV-C. The proposed approach first transfers high-visibility regions through a cut-and-paste operation, followed by selective geometry finetuning. Compared to naive finetuning, this method is not only computationally more efficient but also more robust, as it mitigates degradation in few-shot scenarios by initializing from a better starting point.

First, we utilize the visibility field U in Section IV-B to filter instances with newly visible surfaces due to change (Fig. 5 (a)). We use a volume rendering technique to obtain visibility values from each instance (Fig. 5 (b)). We consider an instance as a target of geometric finetuning if it contains low-visibility surfaces from all new observation viewpoints V. Next, we perform geometric finetuning by leveraging existing geometry $T_i X_i$, the visibility U, and images from V. In order to preserve the certain (well-observed) geometry in $T_i X_i$, we consult U to retain parts with high visibility. Then, with the new observations, we reuse the loss of Section III-B to finetune the geometry as in Fig. 5 (c), resulting in accurate instance-wise geometry.

V. Experiments

In this section, we provide both qualitative and quantitative analysis on the performance of INGRID on identifying individual object instances and their 3D geometries by understanding and utilizing change in the scene.

Implementation-wise, we first predict surface normals and instance masks, each using DSINE [3], and SAM [31]. To reconstruct the objects solely, we retain predicted masks within the predefined workspace of the robot. The workspace is defined to be a 50cm $\times$ 60cm $\times$ 30cm box positioned at the center of the Franka Panda workspace. All of our experiments are conducted on an RTX 3090, with an i7-8700 CPU. In addition to our real-world setup including a Realsense d435i with Franka Panda, we create photo-realistic scenes using Blender Cycles [8] by populating a tabletop with household objects, for groundtruth annotations.

First, we report the qualitative results in Fig. 6. The first column represents images before the change, and the second column visualizes the leaf nodes of the instance candidate tree in different colors. The third column denotes the full hierarchy of the instance candidate tree. The fourth column reports the interactions from our algorithm, while the fifth and last columns each represent the visibility described in Section IV-B and the final instance-wise geometry. The first row originates from our real-world setup, while the second and third row comes from our photo-realistic Blender dataset. For all cases, INGRID successfully captures instance candidates in a tree formulation, estimates, and exploits change to reflect and finetune changes in the geometry.

Second, we provide quantitative results regarding the ability of our model to obtain geometry and instance information. We evaluate on our Blender dataset for accessibility to groundtruth geometry and instance. We render RGB images to 50 viewpoints uniformly sampled from a circle at 45 degrees altitude, from which we retrieve estimates for surface normals and instances using [3, 31].

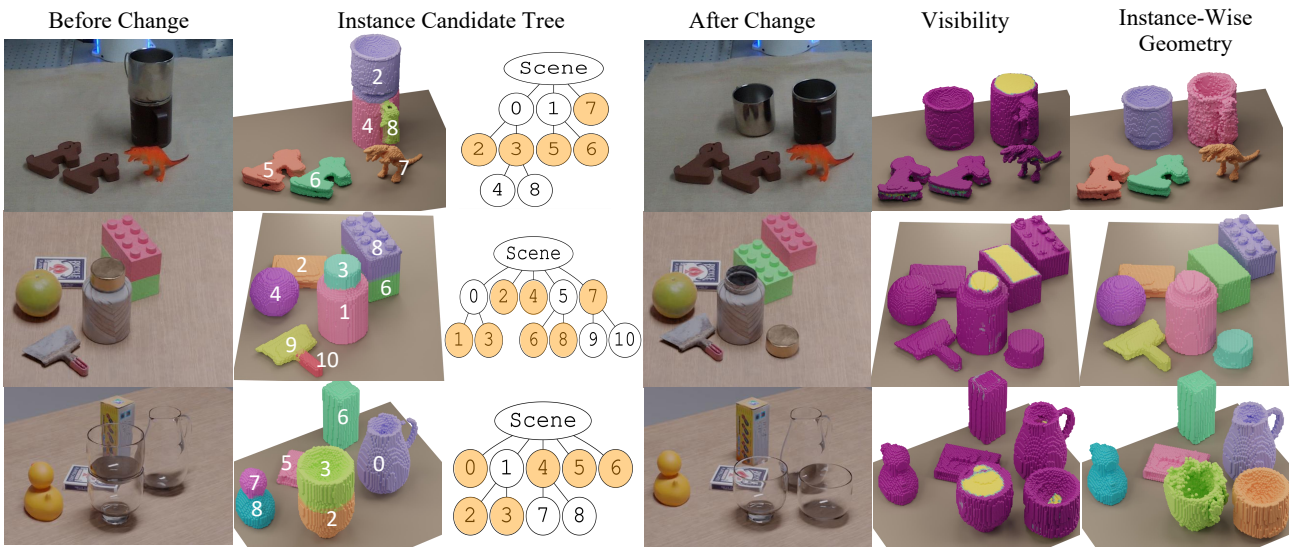


Fig. 6: Instance candidate fields, visibility, and instance-wise geometry for various scenes. Different colors represent different instances in the instance candidate tree and instance-wise geometry. In the instance candidate tree, we only depict the leaf nodes. The yellow region in the visibility map represents a newly visible surface due to the change. INGRID successfully generates instance candidate trees and recovers instance-wise geometry given a change. The recovered instances are shaded in orange in the instance candidate tree.

TABLE I: Geometric update performance of various methods. Metrics include visual surface discrepancy (VSD) of depth [21] and intersection over union (IoU) for 33 novel viewpoints. We report the average metrics over three sets of different viewpoints for each image count. Bold represents the best results while underline refers to the second.

	VSD (↓)				IoU (↑)				Runtime (↓)
Models	2 imgs	4 imgs	8 imgs	16 imgs	2 imgs	4 imgs	8 imgs	16 imgs	Average
INGRID (Update + Finetune + Visibility)	**0.0385**	**0.0330**	**0.0307**	**0.0302**	0.8672	**0.8973**	0.9132	0.9142	17 sec
INGRID (Update + Finetune)	0.0465	0.0413	0.0387	0.0383	0.8673	0.8967	0.9131	0.9134	17 sec
INGRID (Update)	0.0433	0.0416	0.0407	0.0406	**0.8771**	0.8875	0.8937	0.8921	3 sec
INGRID (Scratch)	0.1842	0.0427	0.0343	0.0333	0.5155	0.8894	**0.9550**	**0.9578**	62 sec
Dex-NeRF [23] (Update)	3.0190	2.6412	0.0934	0.0506	0.0769	0.8053	0.9353	0.9461	25 min
Dex-NeRF [23] (Scratch)	2.7979	2.6319	0.0992	0.0535	0.0695	0.4382	0.9351	0.9453	50 min
NeRF [38] (Update)	0.8672	1.0712	0.1003	0.0687	0.0769	0.8053	0.9353	0.9461	25 min
NeRF [38] (Scratch)	1.7426	0.9968	0.0881	0.0716	0.0695	0.4382	0.9351	0.9453	50 min
Instant-NGP [39] (Update)	0.7123	1.1015	0.3884	0.0981	0.5807	0.7565	0.8472	0.8617	25 sec
Instant-NGP [39] (Scratch)	0.3973	0.4515	0.3791	0.0900	0.1617	0.4455	0.7821	0.8665	50 sec

Table I compares the geometric accuracy of variations of our model with other field-based methods such Dex-NeRF [23], NeRF [38], and Instant-NGP [39]. For our method, the update scheme represents instance-wise rigid body transform obtained from Section IV-C, while finetuning refers to geometric finetuning of Section IV-D. Visibility refers to whether visibility-based selective finetuning of Section IV-D is applied. For the update scheme of the baseline models, we follow the method of Evo-NeRF [27] by loading the trained weights and resuming training. We compare in such a manner since our update method cannot be directly applied on continuous field representations such as Dex-NeRF [23], NeRF [38] and Instant-NGP [39].

We evaluate geometry with 2 metrics: Visual Surface Discrepancy (VSD) [21] and Intersection over Union (IoU). VSD refers to the average error of the rendered object depth with respect to the groundtruth depth. IoU measures the match between rendered object masks and groundtruth masks. Since viewpoint selection may drastically change the results, for each number of input images (e.g., 4 imgs), we perform 3 trials on different sets of viewpoints sampled by farthest point sampling on a circle at 45 degrees altitude. For evaluation, we render object depth and mask to 33 novel viewpoints. Our experiments demonstrate that INGRID accurately captures scene geometry under change, while remaining efficient in both computation time and image input. With only 2 images, our method can successfully transform objects in 3D space (update) while finetuning newly visible parts, in 17 seconds.

Table II reports the instance identification performance of our model, the ablated versions of our model, Garfield [29], and OmniSeg3D [53]. The ablated versions (coarse, mid, fine) are given only fixed types of masks from SAM [31], which cannot be updated based on observed changes. Instead, they represent the 3D instances we can obtain from passive visual observation. For Garfield [29], we sweep the scale hyperparameter from 0 to 0.40 in 0.05 steps, but display only from 0 to 0.15, since Garfield [29] outputs only a single cluster for scales over 0.15. For fair comparison, we manually crop the

TABLE II: Precision and recall of 3D bounding boxes. Precision averages IoU over predicted instances, while recall averages over GT instances. Bold represents best results.

Models	Precision (↑)	Recall (↑)
INGRID	**0.7449**	**0.7002**
INGRID (coarse)	0.5707	0.6403
INGRID (mid)	0.6393	0.6816
INGRID (fine)	0.6226	0.5500
Garfield [29] (0.00)	0.5652	0.4218
Garfield [29] (0.05)	0.5268	0.4026
Garfield [29] (0.10)	0.4562	0.4361
Garfield [29] (0.15)	0.3055	0.3107
OmniSeg3D [53]	0.5033	0.5567

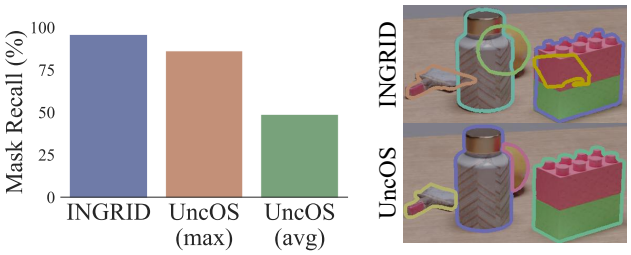


Fig. 7: Average mask recall comparison between INGRID and UncOS [18] (left). For the single-view setting of UncOS, we report both the maximum and average results across viewpoints. We visualize the detected objects for INGRID and UncOS (right).

background geometry captured by Garfield [29] and leave only the objects in the clustering phase, since Garfield [29] cannot take as input background masks. For OmniSeg3D [53], we follow the entire pipeline including the background processing before applying HDBSCAN [37] to obtain 3D clusters.

For the metric, we calculate the precision and recall of the 3D bounding boxes of the predicted instances with respect to the groundtruth bounding box. Precision refers to IoUs of the 3D bounding box averaged over predicted instances while recall refers to IoUs averaged over groundtruth instances. Our experiments report that our method can identify instances more accurately than other methods such as Garfield [29], or OmniSeg3D [53]. In addition, we find that determining instances in an interaction-based manner outperforms utilizing any level of mask predicted via texture from SAM [31].

Finally, we report the results of our experiments comparing INGRID with UncOS [18], which also shares the philosophy of utilizing interaction for segmentation. Fig. 7 compares the mask recall of INGRID and UncOS on our Blender dataset. Recall is measured as the ratio of successfully detected objects, where a detection is considered successful if its IoU exceeds 0.75. For the baseline, we follow UncOS, using both color and depth images from Blender as input. INGRID utilizes RGB images from 16 viewpoints, whereas UncOS operates on a single-view RGB-D input. To ensure fairness, we report both the maximum and average detection performance of UncOS across viewpoints. UncOS exhibited substantial variation between its maximum and average results, highlighting its sensitivity to viewpoint selection. In contrast, INGRID, which can leverage instance and geometry-level information across multiple views, achieved more consistent and robust detection, particularly in occlusion-heavy scenes than the single-viewed UncOS.

VI. Conclusion and Limitations

We proposed INGRID, a method for actively perceiving instance-wise object geometries in occlusion-prone scenes. INGRID (i) estimates and leverages visibility to guide purposeful interactions using a robotic arm, (ii) jointly optimizes instances and their transformations, and (iii) adapts to scene changes while refining geometry with minimal additional data acquisition. Experiments in both real and synthetic environments demonstrate that INGRID reliably reconstructs object instances and their geometries, which can later be utilized to perform object-centric robotic tasks.

While INGRID demonstrates consistent performance across a range of scenes, several limitations remain. First, INGRID assumes that the true object instances are included within the initially constructed instance candidate tree. Although our current approach, which leverages general-purpose segmentation modules [31], was sufficient in our experiments, incorporating mechanisms for dynamic node splitting or merging could enhance the flexibility and robustness of the method. Second, although INGRID effectively recovers object geometries in occluded environments, it relies on sufficient viewpoint coverage around the target objects. As a result, it struggles in scenarios where parts of objects are entirely unobservable, such as items placed on shelves. Integrating prior knowledge (e.g., symmetry) or parametric shape models (e.g., superquadrics) may help to address this limitation. Lastly, the rigid-body assumption limits its applicability to scenes involving deformable objects, such as cloth or dolls. Extending the framework to incorporate a piecewise rigid body model is a promising direction for addressing this constraint.

References

[1] Michal Adamkiewicz et al. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.

[2] Christopher G Atkeson, Chae H An, and John M Hollerbach. Estimation of inertial parameters of manipulator loads and links. *The International Journal of Robotics Research*, 5(3):101–119, 1986.

[3] Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9535–9545, 2024.

[4] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[5] Ruzena Bajcsy and Larry S. Hutchinson. Active vision. *Proceedings of the IEEE*, 76(8):996–1005, 1990. doi: 10.1109/5.5968.

[6] Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos. Active perception: Past, present, and future. *arXiv preprint arXiv:1603.02729*, 2016.

[7] Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi. Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[8] Blender Foundation. Blender - cycles renderer, 2023. URL https://www.blender.org/. Version 3.1,.

[9] Jeannette Bohg et al. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017. doi: 10.1109/TRO.2017.2721939.

[10] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 173–180. IEEE, 2017.

[11] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[12] Jiazhong Cen et al. Segment anything in 3d with nerfs. *Advances in Neural Information Processing Systems*, 36: 25971–25990, 2023.

[13] Haoran Chen, Kenneth Blomqvist, Francesco Milano, and Roland Siegwart. Panoptic vision-language feature fields. *IEEE Robotics and Automation Letters*, 2024.

[14] Xinhua Cheng, Yanmin Wu, Mengxi Jia, Qian Wang, and Jian Zhang. Panoptic compositional feature field for editable scene rendering with network-inferred labels via metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4947–4957, 2023.

[15] Qiyu Dai, Yan Zhu, Yiran Geng, Ciyu Ruan, Jiazhao Zhang, and He Wang. Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1757–1763. IEEE, 2023.

[16] Guoguang Du, Kai Wang, Shiguo Lian, and Kaiyong Zhao. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review*, 54(3):1677–1734, 2021.

[17] Hao-Shu Fang et al. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 39(5):3929–3945, 2023.

[18] Xiaolin Fang, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Embodied uncertainty-aware object segmentation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2639–2646. IEEE, 2024.

[19] Caelan Reed Garret. Pybullet planning. https://pypi.org/project/pybullet-planning/, 2020.

[20] Qiao Gu, Zhaoyang Lv, Duncan Frost, Simon Green, Julian Straub, and Chris Sweeney. Egolifter: Open-world 3d segmentation for egocentric perception. In *European Conference on Computer Vision*, pages 382–400. Springer, 2025.

[21] Tomas Hodan et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[22] Jeffrey Ichnowski*, Yahav Avigal*, Justin Kerr, and Ken Goldberg. Dex-NeRF: Using a neural radiance field to grasp transparent objects. In *Conference on Robot Learning (CoRL)*, 2020.

[23] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dex-nerf: Using a neural radiance field to grasp transparent objects. *arXiv preprint arXiv:2110.14217*, 2021.

[24] Hanxiao Jiang et al. Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation. *arXiv preprint arXiv:2402.15487*, 2024.

[25] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17408–17419, 2023.

[26] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023.

[27] Justin Kerr et al. Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects. In *6th annual conference on robot learning*, 2022.

[28] Ninad Khargonkar, Neil Song, Zesheng Xu, Balakrishnan Prabhakaran, and Yu Xiang. Neuralgrasps: Learning implicit representations for grasps of multiple robotic hands. In *Conference on Robot Learning*, pages 516–526. PMLR, 2023.

[29] Chung Min* Kim, Mingxuan* Wu, Justin* Kerr, Matthew Tancik, Ken Goldberg, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *arXiv*, 2024.

[30] Young Hun Kim, Seungyeon Kim, Yonghyeon Lee, and Frank C Park. T2sqnet: A recognition model for manipulating partially observed transparent tableware objects. In *8th Annual Conference on Robot Learning*.

[31] Alexander Kirillov et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[32] Obin Kwon, Jeongho Park, and Songhwai Oh. Renderable neural radiance map for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9099–9108, June 2023.

[33] Alex X Lee, Henry Lu, Abhishek Gupta, Sergey Levine, and Pieter Abbeel. Learning force-based manipulation of deformable objects from multiple demonstrations. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 177–184. IEEE, 2015.

[34] Junho Lee, Sang Min Kim, Yonghyeon Lee, and Young Min Kim. Nfl: Normal field learning for 6-dof grasping of transparent objects. *IEEE Robotics and Automation Letters*, 2023.

[35] Zhaoshuo Li et al. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023.

[36] Minghuan Liu et al. Manipulation as in simulation: Enabling accurate geometry perception in robots. *arXiv preprint arXiv:2509.02530*, 2025.

[37] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017. doi: 10.21105/joss.00205. URL https://doi.org/10.21105/joss.00205.

[38] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[39] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[40] Adithyavairavan Murali et al. Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training. *arXiv preprint arXiv:2507.13097*, 2025.

[41] Stefan Otte, Johannes Kulick, Marc Toussaint, and Oliver Brock. Entropy-based strategies for physical exploration of the environment's degrees of freedom. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 615–622. IEEE, 2014.

[42] Robert Platt, Leslie Kaelbling, Tomas Lozano-Perez, and Russ Tedrake. Efficient planning in non-gaussian belief spaces and its application to robot grasping. In *Robotics Research: The 15th International Symposium ISRR*, pages 253–269. Springer, 2016.

[43] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[44] Adam Rashid et al. Language embedded radiance fields for zero-shot task-oriented grasping. In *7th Annual Conference on Robot Learning*, 2023.

[45] David Schiebener, Jun Morimoto, Tamim Asfour, and Aleš Ude. Integrating visual perception and manipulation for autonomous learning of object representations. *Adaptive Behavior*, 21(5):328–345, 2013.

[46] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[47] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. In *7th Annual Conference on Robot Learning*, 2023.

[48] Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robotics and Automation Letters*, 5(3):4978–4985, 2020. doi: 10.1109/LRA.2020.3004787.

[49] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021.

[50] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28, 2015.

[51] Tianhao Wu, Chuanxia Zheng, Qianyi Wu, and Tat-Jen Cham. Clusteringsdf: Self-organized neural implicit surfaces for 3d decomposition. In *European Conference on Computer Vision*, pages 255–272. Springer, 2025.

[52] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[53] Haiyang Ying et al. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20612–20622, 2024.

[54] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022.

[55] Runsong Zhu, Shi Qiu, Qianyi Wu, Ka-Hei Hui, Pheng-Ann Heng, and Chi-Wing Fu. Pcf-lift: Panoptic lifting by probabilistic contrastive fusion. In *European Conference on Computer Vision*, pages 92–108. Springer, 2025.

[56] Zihan Zhu et al. Nicer-slam: Neural implicit scene encoding for rgb slam. In *2024 International Conference on 3D Vision (3DV)*, pages 42–52. IEEE, 2024.