

Uncertainty Guided Policy for Active Robotic 3D Reconstruction using Neural Radiance Fields

Soomin Lee*, Le Chen*, Jiahao Wang, Alexander Liniger, Suryansh Kumar[†], Fisher Yu

Abstract—In this paper, we tackle the problem of active robotic 3D reconstruction of an object. In particular, we study how a mobile robot with an arm-held camera can select a favorable number of views to recover an object’s 3D shape efficiently. Contrary to the existing solution to this problem, we leverage the popular neural radiance fields-based object representation, which has recently shown impressive results for various computer vision tasks. However, it is not straightforward to directly reason about an object’s explicit 3D geometric details using such a representation, making the next-best-view selection problem for dense 3D reconstruction challenging. This paper introduces a ray-based volumetric uncertainty estimator, which computes the entropy of the weight distribution of the color samples along each ray of the object’s implicit neural representation. We show that it is possible to infer the uncertainty of the underlying 3D geometry given a novel view with the proposed estimator. We then present a next-best-view selection policy guided by the ray-based volumetric uncertainty in neural radiance fields-based representations. Encouraging experimental results on synthetic and real-world data suggest that the approach presented in this paper can enable a new research direction of using an implicit 3D object representation for the next-best-view problem in robot vision applications, distinguishing our approach from the existing approaches that rely on explicit 3D geometric modeling.

Index Terms—Active 3D reconstruction, robot vision, neural radiance fields, next-best-view selection, uncertainty estimation.

I. INTRODUCTION

Active vision-based robotic 3D reconstruction of an object using images or RGB-D sensors is a vital problem for robot vision and perception [1][2][3]. The primary task of active vision in robotic systems is to skillfully operate the camera pose to capture as much information about the scene as possible. One practical approach to achieve this is by using a robot that can place its visual sensor such that the information gained for a given task is maximized [4][5]. That requires the robotic system to make planning decisions based on its state and current perceptual information of the environment without access to unseen information. This paper tackles the active robot vision problem popularly known as next-best-view determination for object dense 3D reconstruction using multi-view images. A typical active robotic 3D reconstruction method generally consists of three essential steps: *(i)* Given the object’s current information, proposals for the next possible view candidates are defined. *(ii)* The best next view

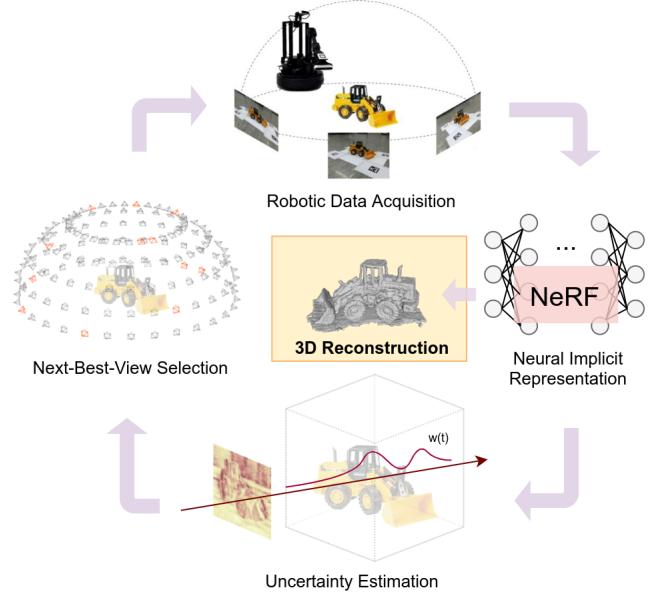


Fig. 1: **Overview.** We develop a robotic system that actively estimates the next-best-view for dense 3D reconstruction of an object leveraging the uncertainty modeling in implicit neural representation. For object representation, our work uses neural radiance fields [11] for its simplicity and notable performance on 3D shape representation.

candidate is selected based on a specified criterion. *(iii)* The robot maneuvers to the corresponding pose to obtain new 3D information about the object. These steps continue until no acceptable information gain is observed (see Fig. 1). Since the robot has no access to the actual candidate views in step *(ii)*, it has to evaluate the unvisited view candidates, making the decision challenging and critical in this pipeline. Thus, the solution of this step is one of the main differentiating factors among existing methods [2][6][7][8][9][10].

Available approaches that estimate the information gain for this problem work on explicit 3D representations such as pointcloud, voxel, etc., which are obtained via structure-from-motion (SfM) or a calibrated RGB-D sensor. However, it is well-studied that SfM has limitations, and its suitability for achieving dense 3D reconstruction for robotic applications remains challenging [12][13][14]. Consequently, methods that rely on the fusion of depth maps coming from an active RGB-D sensor became popular [15][16][17]. Still, such methods are limited by the depth sensor acquisition range, depth sensor noise, and perceived depth accuracy, which is affected by the object’s surface details and material type. Hence, both SfM and RGB-D fusion methods have inherent drawbacks in dense 3D reconstruction, limiting their application in active

*Authors contributed equally.

Soomin Lee is with Oracle Labs Zürich, Switzerland.

Le Chen, Jiahao Wang, Suryansh Kumar, and Fisher Yu are with ETH Zürich, Switzerland.

Alex Liniger is with Huawei Zürich, Switzerland.

Note: This work was completed with VIS Group, ETH Zürich.

[†]Corresponding Author: Suryansh Kumar, ETH (k.sur46@gmail.com).

3D reconstruction.

Recent advances in shape representation based on neural radiance fields, particularly NeRF [11], have shown promising results for several computer vision tasks. Using well-posed multi-view images, NeRF can provide an object's dense 3D reconstruction with favorable accuracy, overcoming the inherent limitations with SfM [13][14][18] and depth fusion methods [15][17]. Accordingly, we leverage the NeRF representation for the active robotic 3D reconstruction task. Contrary to the available methods that rely on explicit 3D representations, we explore the possibility of implicit neural shape representation to solve this problem. Nevertheless, due to the implicit nature of NeRF, the estimation of the information gain becomes even more challenging. We show that by computing the entropy of the weight distribution in the NeRF representation, we can reason about the information gain. We demonstrate with examples that such an approach is possible and can be effective.

Contributions. To summarize, our key contributions are:

- We introduce a new method of using implicit neural shape representation for the active robotic 3D reconstruction task.
- We show that entropy of the weight distribution of the color samples can be a suitable proxy for the uncertainty of the underlying 3D geometry, and we present an uncertainty guided policy using NeRF representation for next-best-view selection.

We provide extensive evaluation and comparison results on synthetic benchmark datasets to show the strength of our approach. Our experiments confirm the transferability of the proposed approach to real scenarios with superior results compared to the competing baselines.

II. RELATED WORK

Active robotic 3D reconstruction. Available methods for solving this task generally rely on explicit 3D representations of the object. Isler *et al.*[2] addresses information gain formulation in volumetric representations and compares the proposed metric with different methods in [3]. Although the metric can be effective, it is a combination of several hand-crafted factors. Several other methods use point cloud for object representation, which is another widely used explicit 3D representation. Wu *et al.*[10] uses Poisson fields to get a confidence map of the current estimate to decide the part of an object that needs further scanning. The method focuses on the quality and accuracy of the recovered 3D surface, compromising on runtime instead. Wu *et al.* replace the Poisson fields-based analysis with point completion network [19] to find incomplete parts of a scan, boosting up the speed but limiting their attention to plant phenotyping [9].

In terms of robotic platforms that enable active 3D reconstruction, [2] has the most similar setting as ours. They use a wheeled mobile robot to move around an object and scan it with a camera. [9] and [10] both demonstrate their ideas on a robot but with a fixed base, using a scanner and an RGB-D sensor with multiple robot arms, respectively.

Neural 3D shape representations. Neural implicit shape representation via multi-layer perceptron (MLP) has recently gained popularity as an effective representation for 3D shapes [11][20][21]. Neural implicit representations are independent of spatial resolution as geometry can be represented continuously without discretization and has a lower memory footprint. Earlier works for such a representation optimize a network to regress either the Signed Distance Function (SDF) or the occupancy function that takes 3D coordinates as an input [22][23]. Although these methods can successfully represent 3D shapes, they require 3D supervision, restricting their use to problems where the 3D geometry is unknown.

To our knowledge, this paper is an early attempt to build an active robotic 3D acquisition system based on a neural implicit representation of an object. While it is not yet common to adopt neural representations in robotics applications due to time constraints, a recent work [24] succeeded in using a neural representation to represent scenes in a real-time system. Furthermore, they demonstrated for the first time that a multi-layer perceptron could serve as the scene representation for an RGB-D SLAM system.

Volume and surface rendering for 3D reconstruction. In the past, SfM, multi-view stereo, and depth-map fusion-based methods were widely used for active 3D acquisition tasks [2][13][14][15][25]. However, as alluded to above, these classical approaches have limitations in the dense 3D reconstruction of an object.

Recently, neural volume and surface rendering methods have shown excellent 3D object reconstruction results. For instance, DVR [26] introduced a differentiable volumetric rendering formulation for multi-view 3D reconstruction using image data only. On the contrary, IDR [27] introduced a surface reconstruction approach leveraging a neural renderer that approximates the light reflected from the surface. Other recent approaches leverages implicit neural surfaces representation together with volume rendering idea for better 3D reconstruction [28][29][30]. Nevertheless, among all, NeRF [11] turns out to be one of the most popular and widely used volume rendering approaches for object dense 3D acquisition and novel view-synthesis tasks.

NeRF is a simple yet effective volume rendering approach. It represents the continuous static scene as 5D neural radiance fields, parameterized by multi-layer perceptron (MLP). It demonstrated that regressing density and light fields via an MLP could yield photo-realistic rendering. Due to its remarkable ability to capture complex geometric details, it gathered significant interest from the community. NeRF led to several recent follow-up works that try to reduce the training time [31][32], handling unknown or noisy camera poses [33][34], adding depth supervision [35], or adding a notion of uncertainty [36] [37]. Since NeRF is simple, powerful, and forms the basis of all recent neural rendering works mentioned above, we choose NeRF methodology for this paper. Consequently, the idea presented in this paper can generalize to various other representations that stem from NeRF.

III. METHOD

Our work puts forward a policy formulation that selects the best candidate views for improving the 3D reconstruction in an active robot setting. Our approach infers the uncertainty from a proposed pose by synthesizing the novel view from NeRF-based shape representation. The rest of the section is organized as follows: Sec. III-A provides an overview of NeRF formulation [11]. Next, Sec. III-B introduces our approach to model uncertainty. Lastly, Sec. III-C, describes our formulation for the uncertainty guided policy.

A. Preliminaries

NeRF [11] models the continuous radiance fields of a static scene using a multilayer perceptron (MLP). It takes a set of images and encodes the scene as a volume density (σ) and color $\mathbf{c} = (r, g, b)$. NeRF renders each pixel of an image in a following way: Given a 3D point (x, y, z) in the scene space and the ray direction parameterized by (θ, ϕ) emitted from the camera's center of projection \mathbf{o} , NeRF learns an implicit function that approximates the scene σ and \mathbf{c} via an $\text{MLP}((x, y, z, \theta, \phi); \Theta) = (\sigma, \mathbf{c})$, where Θ is the parameter of an MLP network. Using σ and \mathbf{c} per scene point, we can render images from novel views via volume rendering [38].

Consider a ray r emanating from a camera position $\mathbf{o} \in \mathbb{R}^3$ in direction $\mathbf{d} \in \mathbb{R}^3$, where $\|\mathbf{d}\| = 1$. Volume rendering approximates light radiance by integrating the radiance along the ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}, t \geq 0$. Specifically, the expected color $\mathbf{C}(\mathbf{r})$ is computed using

$$\mathbf{C}(\mathbf{r}) = \int_0^\infty T(t) \sigma(\mathbf{r}(t)) \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt, \quad (1)$$

where $T(t)$ indicates the accumulated transmittance along the ray r and is defined as

$$T(t) = \exp\left(-\int_0^t \sigma(\mathbf{r}(s)) ds\right). \quad (2)$$

It can be interpreted as the probability that a light particle traverses the segment $[\mathbf{o}, \mathbf{r}(t)]$ without being bounced off. It's complement probability, denoted as the opacity O , is defined as $O(t) = 1 - T(t)$. The opacity O is a monotonic increasing function with $O(0) = 0, O(\infty) = 1$. Thus, the opacity function O can be regarded as a cumulative distribution function, and its derivative as a probability density function (PDF) [30]

$$w(t) = \frac{dO}{dt}(t) = T(t) \sigma(\mathbf{r}(t)) \quad (3)$$

The integrals in Eq.(1) and Eq.(2) can be estimated numerically using quadrature approximation [39] as,

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i \quad (4)$$

where, $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ and $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples. $\hat{\mathbf{C}}(\mathbf{r})$ in Eq.(4) can be viewed as a weighted sum of color samples c_i , and can be written as

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^N w_i \mathbf{c}_i, \text{ where, } w_i = T_i (1 - \exp(-\sigma_i \delta_i)) \quad (5)$$

The weight term w_i in Eq.(5) is a discrete approximation of the continuous weight expression in Eq.(3), and can be derived

using the approximation $\sigma_i \delta_i \approx (1 - \exp(-\sigma_i \delta_i))$. Furthermore, if we define $p_i = \exp(-\sigma_i \delta_i)$, the discretized weight can be expressed as follows:

$$w_i = (1 - p_i) \prod_{j=1}^{i-1} p_j \quad (6)$$

B. Ray-Based Volumetric Uncertainty

As mentioned earlier, the neural radiance field representation has several advantages over other shape representations. At the same time, it can provide dense 3D reconstruction using multiview images only. Yet, due to the implicit nature of the representation, it is not straightforward to directly operate on the explicit 3D shape and evaluate the 3D shape that the current network will yield. Moreover, the reasoning about the correctness of the object's volume density pivot around the multiview RGB color rendering values, making the inference about the 3D shape rather challenging.

We address such a challenge in potential next view selection by analyzing the distribution of weight, $w(t)$ in Eq.(3), along the rays of each pixel. Assuming we are looking for a solid surface, an ideal model should have a concentrated weight around the surface and nowhere else. This is also theoretically motivated since the weight term can be regarded as the derivative of the opacity, as shown in Eq.(3). Thus, the weight distribution will have one clear peak if the network correctly learns about the surface. [30] showed that the weight distribution indeed gets closer to a step function at the surface during training. We use the same observation to determine regions where NeRF has not yet learned a sufficiently good 3D representation. We argue that the regions with non-concentrated weights are where the 3D geometry can be improved.

To confirm our hypothesis, we study distributions of weight $w(t)$ along rays as shown in Fig.(2). The first distribution shows a ray that intersects with a relatively accurate part, and has a clear peak. The second distribution shows a ray that intersects with a noisy part, which has a noisy peak, but it is still concentrated. Finally, the third distribution shows a ray that intersects with an incomplete part, and the distribution has multiple peaks and is spread out. These results coincide well with our hypothesis. In summary, by examining how concentrated the weight distribution is, we can infer how certain the network is about the ray.

Specifically, we quantify the degree of how concentrated a distribution is with entropy. Given a discrete random variable X , the entropy of X is defined as:

$$H(X) = -\sum_{i=1}^n P(x_i) \log P(x_i), \quad (7)$$

where $P(x_i)$ denotes the probability of the event $X = x_i$. Entropy fits our purpose because evaluating whether a distribution has one sharp peak is in consonance with evaluating the uncertainty of a random variable the distribution yields. A uniform probability distribution yields the maximum entropy, while the entropy becomes zero when the outcome is always determined. Note that the weight $w(t)$ can be viewed as a PDF, as discussed earlier, so the definition of entropy can indeed be applied to the weight distributions.

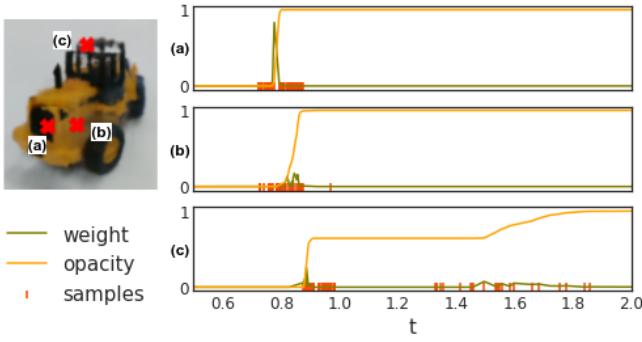


Fig. 2: **Key observation for the proposed ray-based volumetric uncertainty.** The weight $w(t)$, opacity $O(t)$, and sampled positions along rays are visualized. We can infer the uncertainty of the underlying reconstruction of the object by analyzing the weight distribution. (a) Accurate part: concentrated weight distribution with a clear peak. (b) Noisy part: concentrated weight distribution with a noisy peak. (c) Incomplete part: spread out distribution with multiple peaks.

Fig.(3) shows whether the uncertain regions align well with the inaccurate recovery of a 3D mesh. We collect 60 images along a single horizontal circular trajectory around a toy loader using our robotic system and we train a model using those images. Then a 3D mesh is extracted from the model, which contains inaccurate regions due to insufficient coverage of the object in the training data. One can confirm that the parts with high entropy match well with those not precisely reconstructed, such as high-frequency regions.

Our proposed uncertainty estimation has several advantages: First of all, the idea can be directly generalized to different works that are based on neural rendering. Estimating the weight distribution along each ray is one of the processes that commonly exist in every work that leverages neural rendering. Next, it is simple and easily applicable because it does not require any additional training or changes in the network. Finally, it provides a metric that can be evaluated on the combined effects of different sources of uncertainty, such as deficiency of data or geometric complexity. Accordingly, we can avoid reasoning about different criteria we need to consider, eliminating the need to use heuristics as in [2].

Several recent works proposed methods to identify the uncertainty in NeRF [36][37]. NeRF-W [36] models static and transient elements separately in order to handle uncontrolled images, and the notion of uncertainty mainly serves as an attenuation factor for the transient elements rather than the uncertainty of 3D reconstructions of static scenes. S-NeRF [37] learns to encode the posterior distribution over all the possible radiance fields modeling the scene and obtains the uncertainty estimates by sampling, following a Bayesian approach. However, both of them require models to be modified, while our method can be directly plugged into other works that are based on neural rendering.

Note: The proposed approach, however, is restricted to non-transparent objects with solid surfaces. Volume rendering-based models such as NeRF can model transparency. Yet, it is not straightforward to recover the 3D surface of the transparent glass. The extraction of 3D geometry usually relies on the Marching Cube [40] algorithm, which can easily remove low volume density regions together with noise. Hence, our work is suitable for non-transparent objects.

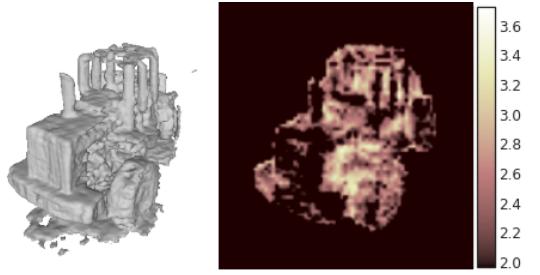


Fig. 3: **Correlation between 3D mesh and entropy map.** Less precise parts in the 3D mesh (left) coincide with higher entropy parts in the object-masked entropy map (right). Brighter pixels indicate higher entropy values.

C. Uncertainty Guided Policy

For efficient active robotic 3D reconstruction, the fundamental task is to decide which views to scan next. As highlighted in the introduction, the challenge comes from having no access to the view at the decision time, i.e., step *(ii)*. Our proposed ray-based volumetric uncertainty estimation approach allows us to infer the importance of adding a novel view image via its uncertainty estimate to address such a challenge. Thus, by design, it is straightforward to convert the ray-based volumetric uncertainty estimator into a policy by considering the mean entropy of a candidate view as a proxy to the information gain this view can bring.

When we select images based on the proposed implicit volumetric uncertainty, we take the mean of the entropy values across all pixels to be the representative value for an image. We find the mean values sufficient for our task of view selection; however, one can potentially reason about local uncertainties using the information since we compute the uncertainty measure for each pixel.

In this work, we investigate a coarse-to-fine reconstruction approach, where we start with a coarse set of images and select views to improve the initial reconstruction. Consequently, we introduce region clustering, as shown in Fig.(4), which consists of a region where the initial views are selected and several additional regions where further views can be selected. After a coarse reconstruction using the initial views, each iteration selects the view with the highest mean entropy of each sector, and the robot collects the corresponding view. These views are then added to the dataset, based on which the model is refined. Without region clustering, the overall acquisition process will be much longer to cover the object geometry from different viewpoints, as we will get only one view every iteration. Alternatively, selecting the top-k most uncertain viewpoints without updating the model may result in choosing a group of similar viewpoints. However, this is not optimal since only a subset of them may be sufficient to reduce the uncertainty in the region, and using similar views could lead to overfitting. On the other hand, splitting the view selection space into regions is a simple yet effective step that helps select diverse views, which can potentially be used to incorporate a prior based on domain knowledge. We discuss other selection policies in Sec. IV-D.

IV. EXPERIMENT

We evaluate our uncertainty guided policy on both synthetic and real-world data covering several types of objects. For

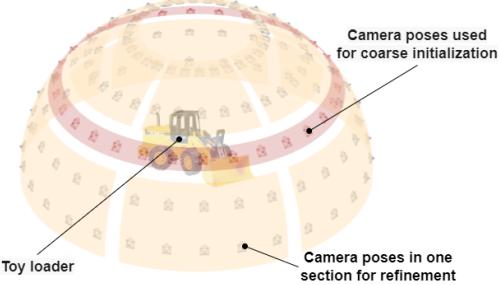


Fig. 4: **Region clustering.** The view space defined on a hemisphere is divided into several sections to locally determine additional training samples. The middle part indicates the circular trajectory where the initial poses are sampled from, and thus is excluded when clustering camera poses for additional training.

clarity, experimental setup and results on synthetic and real-world dataset are described separately in Sec. IV-A and Sec. IV-B, respectively.

Implementation Details. We define the view space to be a hemisphere surrounding the object and acquire images of the object from five circles with different radii on the hemisphere (see Fig.(4)). Thirty candidate poses are defined for each of the five horizontal circles on the hemisphere, resulting in 150 camera poses. Then as an initialization, we train a NeRF model using images only from the middle circle. We use six images for initialization (coarse reconstruction) in experiments with synthetic data, while we use 15 images for real-world experiments. We use this setup as a few images are enough to get a high-quality 3D mesh for synthetic data. Still, significantly more images are required for real objects, showing the importance of both synthetic and real-world experiments. Then we divide the hemisphere into 12 sections, as shown in Fig.(4) for region clustering. The hemisphere is divided into the upper and lower half with respect to the middle circle that contains the initial training images, and each half is further divided into six groups according to their azimuthal angles. Therefore, in total, we have 12 sections and one middle circle, and we select 12 additional images , one from each section in each iteration.

For training a NeRF model, we use the official code provided by the authors [11]. We use 64 samples for the ‘coarse’ network and 128 samples for the ‘fine’ network. When evaluating the entropy of the weight distributions, we downsample the images with a factor of 4 to speed up the process. After selecting additional images, we initialize the network with the model from the initialization step and refine the model further with the updated training set.

Evaluation Metric. For 3D reconstruction evaluation, we use the popular F-score metric [41]. The F-score is the harmonic mean of precision and recall at a certain threshold d . The precision quantifies the accuracy of the reconstruction, and it can be maximized by producing a very sparse set of precisely localized landmarks for instance. The recall quantifies the completeness of the reconstruction, and it can be maximized by densely covering the space with points.

Hardware. We use LoCoBot, a low-cost mobile manipulator hardware platform to perform active robotic 3D reconstruction

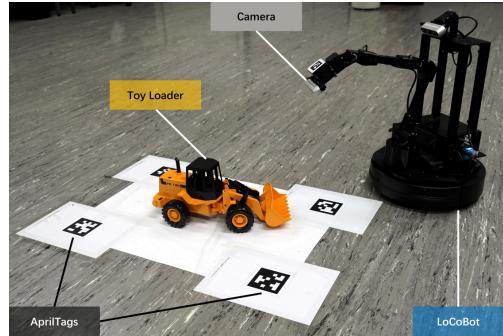


Fig. 5: **Experimental setup for real-world object reconstruction.** The toy loader is placed in the middle of 4 AprilTags [43] for localization. The gripper of LoCoBot [42] is replaced with a camera.

[42]. It has a wheeled mobile base with two degrees of freedom (DoF), a manipulator with 5 DoF, and a camera attached to the top. Initially, the robot had a gripper at the end of the manipulator, but it was replaced with another camera to allow the robot to perform exact 3D reconstruction. Note that we *do not* use depth information from the RGB-D sensor. The adjusted hardware and the experimental setup are shown in Fig.(5). Using the camera on top, we localize the robot with respect to AprilTags [43] to control the robot and to position the camera on the arm to acquire well-posed images.

A. 3D Reconstruction of Synthetic Objects

(a) Datasets. For this experiment, we use the NeRF Blender dataset [11]. We select four objects, namely *Lego*, *Chair*, *Drums*, and *Ficus* to generate dataset for our robotic setup. Note that the transparent surfaces of the *Drums* model are removed to satisfy our assumption. We render 150 images according to our experimental setup mentioned before.

(b) Baselines. For each object, we report the results of the initial model trained with 6 images from the middle circle and the model trained with all 150 images from the hemisphere. We also present 5 different next-best-view selection policies as baselines. *(i)* *Random policy*: a pose is selected randomly within each section. *(ii)* *Heuristic policy*: the middle pose of each of the 12 sections is selected. *(iii)* *Similarity policy*: within each section, a pose with the lowest image similarity to the initial training data is selected. Using the initial NeRF model, we synthesize candidate views and compute the cosine similarity between the feature vector of the synthesized views and the initial training images. The feature vector is obtained with ResNet-18 [44] pre-trained on ImageNet [45]. *(iv)* *Similarity (GT) policy*: a pose is selected in the same way as *(iii)*, but the ground truth images are used for feature extraction instead of the synthesized views. While the baselines *(i)-(iv)* all are based on NeRF, we additionally compare our approach with a volumetric active 3D reconstruction method, denoted as *(v)* *Volumetric Information (VI) policy* [2]. Note that this method uses stereo images as input. *(vi)* *Pure Random*: together with the aforementioned policies, we also present a pure random baseline where images are randomly chosen over the entire view space rather than within each section.

(c) Results. We run each baseline view selection policy based on the initial model trained with 6 images to select one image within each section. It means that after one iteration, we have

TABLE I: **F-score of synthetic object reconstruction.** With access to the ground truth meshes, we show a quantitative comparison against the baselines for the reconstructed geometry on 4 different synthetic objects. Our method performs the best among all the selection policies. We also report the results of the mesh reconstructed with all 150 images for reference. Bold numbers are only considering policies. The computation time for the next-best-view selection using each policy: Random and Heuristic- less than a second, Similarity (GT)- 15 sec., VI[2]- 13.8 sec., Similarity and Ours- approx. 5 min.

Object	Initialization	Pure Random	Policy						All Images
			Random	Heuristic	Similarity	Similarity (GT)	VI [2]	Ours	
Lego	0.3549	0.3682	0.3909	0.3959	0.3873	0.3710	0.1824	0.4101	0.4374
Chair	0.1285	0.1696	0.1831	0.1615	0.1772	0.1836	0.0959	0.1858	0.2142
Drums	0.2778	0.2229	0.2766	0.2700	0.2732	0.2687	0.1548	0.2853	0.3793
Ficus	0.1788	0.2557	0.2622	0.2666	0.2676	0.2630	0.1735	0.2705	0.3781

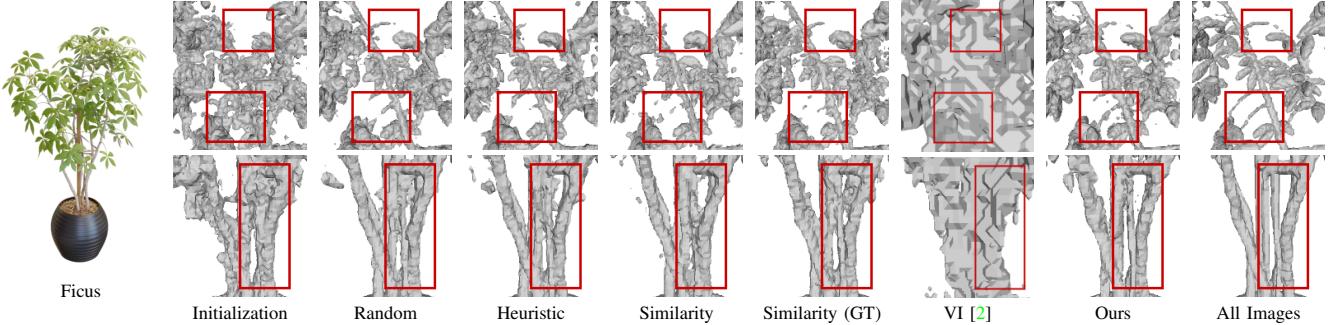


Fig. 6: **Comparisons on 3D meshes for Ficus.** We show a qualitative comparison against the baselines for the reconstructed geometry on *Ficus*, which has high frequency details. Our method captures the fine geometry well compared to the other policies. Note that the VI [2] method yields voxel representations.

12 additional images to refine the reconstruction. We report the reconstruction results after one iteration of different view selection policies in Table I. From Table I, we can see that for all the synthetic objects we have tested on, our uncertainty guided policy obtains the highest F-score against all the baselines. Note that all NeRF-based baseline policies, except for the pure random baseline (i.e., (vi)), are relatively similar since they all get 18 views as an input which are reasonably well distributed. Further, our method achieves improvement up to 30% compared to the pure random baseline. On the whole, these statistical results demonstrate that our policy selects the best view from each region on the hemisphere.

Additionally, the visual similarity policy study shows that the ray rendering uncertainty is more informative than the visual features. Moreover, VI [2] baseline by far achieves the lowest quality reconstruction. This affirms the suitability of our choice of using implicit neural implicit volumetric representation and modern continuous view-synthesis approaches like NeRF to solve this problem. Fig.(6) shows the qualitative results compared to the defined baselines. Clearly, the reconstructions from our method better capture the fine geometric details and coherent overall global shape.

B. 3D Reconstruction of Real-World Objects

(a) Datasets. We use our robotic system to acquire images of real-world objects. For this experiment, we used the toy loader shown in Fig.7a as the target object and acquired images at 640×480 resolution. We compute the camera poses using COLMAP [13][25]. Similar to the synthetic data experiment, we define 150 candidate camera poses. The robot takes about 1.2 minutes to collect 15 images from the middle circle to initialize the coarse 3D shape representation. It takes another 1 minute to collect 12 more images for refinement using the proposed uncertainty guided policy.

(b) Baselines. We reconstruct the toy loader under two baseline settings: **(I)** COLMAP and screened Poisson surface reconstruction [46], **(II)** a reconstruction algorithm provided by Open3D [16][47] that uses RGB-D image sequences. For baseline **(I)**, we use all 150 images for reconstruction. When we use the same 27 images that are used for our method, the structure-from-motion pipeline fails because the images contain large rotational changes in their views. For baseline **(II)**, an RGB-D image sequence is required and hence an RGB-D image sequence consisting of 2000 images is used.

(c) Results. The resulting 3D meshes are shown in Fig.(7). Conceivably, 3D geometry recovered using our method has the highest quality with fewer images. It can be observed the COLMAP fails to reconstruct the fine surface geometric details (Baseline **(I)**), whereas the RGB-D method provides overly smooth surface reconstruction (Baseline **(II)**). On the contrary, our method can reconstruct the fine surface details while maintaining the global shape structure. Such results validate our idea of uncertainty guided policy for active 3D data acquisition using neural rendering principle.

C. Discussion on Runtime

The average initial training time of the NeRF [11] is around 15 hours when trained on a single NVIDIA GPU (TITAN Xp). Although current experiments cost a significant amount of time to train NeRF models, our uncertainty estimation is general and can be adopted in various NeRF-based approaches. To this end, we performed additional experiments using TensoRF [32], a NeRF-based model that achieves fast training. The training time is about 8.2 minutes on a single NVIDIA 2080 GPU, and the F-scores from different policies when evaluated on *Lego* are: Initialization (0.3272); Random (0.4895), Heuristic (0.4234), Similarity (0.4163), GT similarity (0.4598), Ours (0.5078); All Image (0.6281). Thus, our policy is model

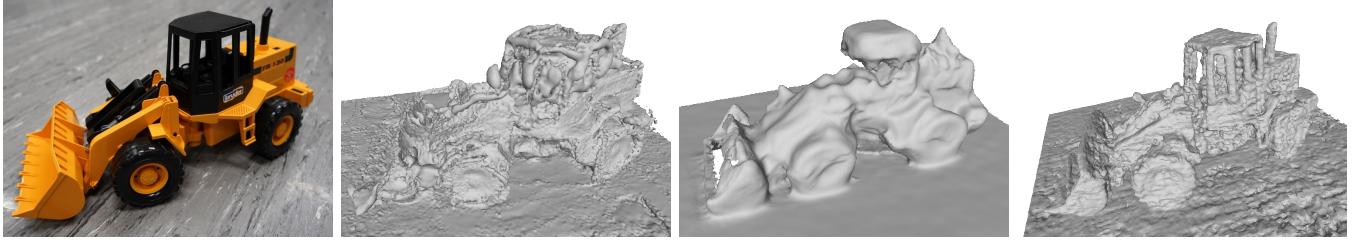


Fig. 7: **Reconstruction results of a toy loader using classical methods and our approach with real-world data.** The toy loader is reconstructed using COLMAP [13], [25] with 150 images, the reconstruction system provided by Open3D [16], [47] with 2000 images, and our approach with 27 images (15 for initialization, 12 for refinement).

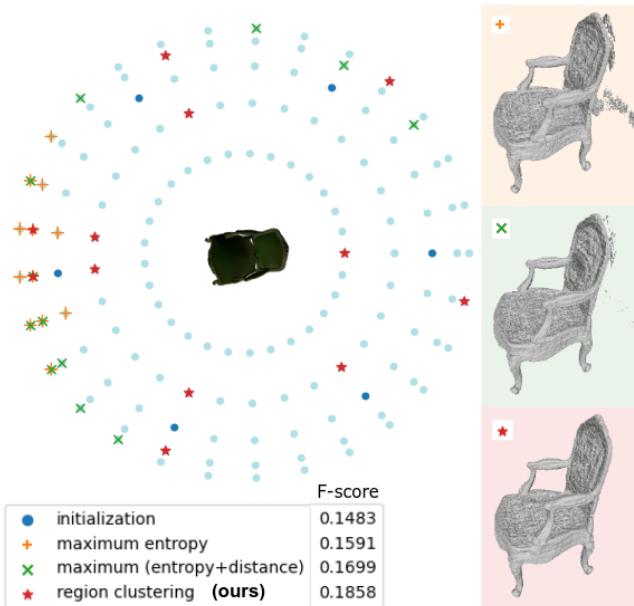


Fig. 8: **Ablation study on view selection policies.** The left plot shows the selected poses using each policy, seen from above the hemisphere. Without the region clustering we proposed in Sec. III-C, selected poses have similar viewpoints with each other, resulting in less information gain and thus less precise reconstruction.

agnostic, and the proposed idea can be switched to an alternative NeRF-based model depending on the application. In addition, we report the computation time for the next-best-view selection using each policy: Random and Heuristic - less than a second, Similarity (GT) - 15 sec., VI [2] - 13.8 sec., Similarity and Ours - approximately 5 min. Further, a faster inference time can be achieved with TensoRF [32], reducing the uncertainty computation time from 5 minutes to 2.8 minutes.

D. Ablation Study

(a) View Selection Policy. As discussed in Sec. III-C, we use region clustering to avoid selecting the next views concentrated in a certain region. We show the proposed policy is an effective approach by comparing the reconstruction result to (i) when we select the most uncertain views without considering other factors such as the locality (+ symbol in Fig.(8)), and (ii) when we consider the average spherical distances between each pair of chosen poses (x symbol in Fig.(8)). Fig.(8) shows the chosen next views for each policy and the resulting 3D meshes. If we use policy (i), all 12 selected views are next to each other and none of them contains the side or the rear view of the object. Therefore, the rear part of the reconstruction

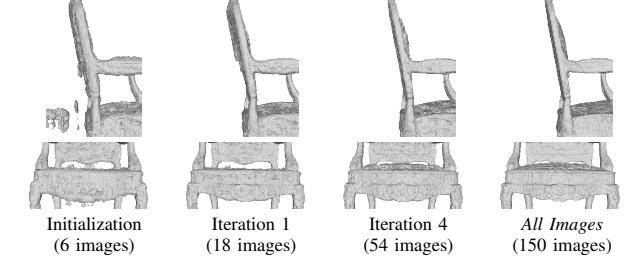


Fig. 9: **Iterative reconstruction result.** After 4 iterations using our uncertainty guided policy, the model is trained using only 54 images, but the mesh quality of the model is comparable to when all 150 images are used.

is particularly noisy. When policy (ii) is used, the selected views are more distributed than in the former case but the rear part of the object is not yet precisely reconstructed. On the contrary, when using our uncertainty guided policy (★ symbol in Fig.(8)), denoted as region clustering, we recover 3D mesh with better quality.

(b) Iterative Reconstruction. When we run our active robotic 3D reconstruction pipeline, we show that we can achieve a comparable level of 3D reconstruction quality without using the whole image set. Fig.(9) shows the resulting 3D meshes of *Chair* at different iterations. After one iteration, the noise floating behind the chair is filtered; however, the rear part of the backrest is still noisy, and the subtle convex shape of the seat and the front part of the backrest is not captured. After four iterations, these fine details are well represented, resulting in a mesh similar to the one we can get by using 150 images with only 54 images. In addition, we observed a decrease in the average of the mean entropy values over all pixels after each iteration: 1.748, 0.837, 0.797, 0.791, 0.790, which complies with our intuition. Therefore, we can efficiently reconstruct an object by actively choosing the camera poses with our policy.

V. CONCLUSION

This paper leveraged neural radiance fields-based implicit representations to tackle active robotic 3D reconstruction of an object. First, we introduced the ray-based volumetric uncertainty estimator, which provides a suitable proxy for the uncertainty of the underlying 3D geometry by computing the entropy of the weight distribution of color samples along rays. The proposed uncertainty estimation is applicable and can be generalized to other recently improved neural rendering-based approaches. Then, based on the estimator, we proposed an uncertainty-guided policy for the robotic system to determine the next best view for effective 3D reconstruction of an object.

Experiments on synthetic and real-world examples show that our policy selects informative views for the object’s better active 3D reconstruction.

Indeed, the optimization and rendering time of classical NeRF can be argued. Nevertheless, as tested using TensoRF [32] implementation, our method is general and can be further improved using advanced neural rendering methods with faster implementations. We believe our proposed method opens up a new research direction of using an implicit 3D object representation for the next-best-view selection problem in robot vision applications.

REFERENCES

- [1] D. Peralta, J. Casimiro, A. M. Nilles, J. A. Aguilar, R. Atienza, and R. Cajote, “Next-best view policy for 3d reconstruction,” in *ECCV*. Springer, 2020, pp. 558–573.
- [2] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, “An information gain formulation for active volumetric 3d reconstruction,” in *ICRA*, 2016, pp. 3477–3484.
- [3] 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.
- [4] S. Chen, Y. Li, and N. Kwok, “Active vision in robotic systems: A survey of recent developments,” *The International Journal of Robotics Research*, vol. 30, pp. 1343 – 1377, 2011.
- [5] 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*, pp. 1–21, 2020.
- [6] M. Devrim Kaba, M. Gokhan Uzunbas, and S. Nam Lim, “A reinforcement learning approach to the view planning problem,” in *CVPR*, 2017, pp. 6933–6941.
- [7] 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.
- [8] Y. Wang, S. James, E. K. Stathopoulou, C. Beltrán-González, Y. Konishi, and A. Del Bue, “Autonomous 3-d reconstruction, mapping, and exploration of indoor environments with a robotic arm,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3340–3347, 2019.
- [9] C. Wu, R. Zeng, J. Pan, C. C. Wang, and Y.-J. Liu, “Plant phenotyping by deep-learning-based planner for multi-robots,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3113–3120, 2019.
- [10] S. Wu, W. Sun, P. Long, H. Huang, D. Cohen-Or, M. Gong, O. Deussen, and B. Chen, “Quality-driven poisson-guided autoscanning,” *ACM Trans. Graph.*, vol. 33, no. 6, Nov. 2014.
- [11] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF: Representing scenes as neural radiance fields for view synthesis,” in *ECCV*, 2020.
- [12] S. Bianco, G. Ciocca, and D. Marelli, “Evaluating the performance of structure from motion pipelines,” *Journal of Imaging*, vol. 4, no. 8, p. 98, 2018.
- [13] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *CVPR*, 2016.
- [14] Y. Furukawa and J. Ponce, “Accurate, dense, and robust multiview stereopsis,” *IEEE T-PAMI*, vol. 32, no. 8, pp. 1362–1376, 2009.
- [15] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.
- [16] S. Choi, Q.-Y. Zhou, and V. Koltun, “Robust reconstruction of indoor scenes,” in *CVPR*, 2015, pp. 5556–5565.
- [17] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in *2011 10th IEEE international symposium on mixed and augmented reality*. IEEE, 2011, pp. 127–136.
- [18] C. Wu *et al.*, “Visualsfrm: A visual structure from motion system,” 2011.
- [19] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, “Pcn: Point completion network,” in *3DV*, 2018.
- [20] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, “Implicit geometric regularization for learning shapes,” in *Proceedings of Machine Learning and Systems 2020*, 2020, pp. 3569–3579.
- [21] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler, “Neural geometric level of detail: Real-time rendering with implicit 3d shapes,” in *CVPR*, 2021, pp. 11358–11367.
- [22] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3d reconstruction in function space,” in *CVPR*, 2019, pp. 4460–4470.
- [23] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “Deepsdf: Learning continuous signed distance functions for shape representation,” in *CVPR*, 2019, pp. 165–174.
- [24] E. Sucar, S. Liu, J. Ortiz, and A. Davison, “iMAP: Implicit mapping and positioning in real-time,” in *ICCV*, 2021.
- [25] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, “Pixelwise view selection for unstructured multi-view stereo,” in *ECCV*, 2016.
- [26] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, “Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision,” in *CVPR*, 2020, pp. 3504–3515.
- [27] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, “Multiview neural surface reconstruction by disentangling geometry and appearance,” *NeurIPS*, vol. 33, 2020.
- [28] M. Oechsle, S. Peng, and A. Geiger, “Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction,” in *ICCV*, 2021, pp. 5589–5599.
- [29] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, “Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction,” *NeurIPS*, 2021.
- [30] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, “Volume rendering of neural implicit surfaces,” *NeurIPS*, vol. 34, 2021.
- [31] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “pixelnerf: Neural radiance fields from one or few images,” in *CVPR*, 2021, pp. 4578–4587.
- [32] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, “Tensorf: Tensorial radiance fields,” *arXiv preprint arXiv:2203.09517*, 2022.
- [33] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, “NeRF--: Neural radiance fields without known camera parameters,” *arXiv preprint arXiv:2102.07064*, 2021.
- [34] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, “Barf: Bundle-adjusting neural radiance fields,” in *ICCV*, 2021.
- [35] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, “Depth-supervised NeRF: Fewer views and faster training for free,” in *CVPR*, June 2022.
- [36] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, “Nerf in the wild: Neural radiance fields for unconstrained photo collections,” in *CVPR*, 2021, pp. 7210–7219.
- [37] J. Shen, A. Ruiz, A. Agudo, and F. Moreno-Noguer, “Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations,” in *3DV*. IEEE, 2021, pp. 972–981.
- [38] J. T. Kajiya and B. P. Von Herzen, “Ray tracing volume densities,” *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 165–174, 1984.
- [39] N. Max, “Optical models for direct volume rendering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.
- [40] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’87. New York, NY, USA: Association for Computing Machinery, 1987, p. 163–169.
- [41] A. Knapsch, J. Park, Q.-Y. Zhou, and V. Koltun, “Tanks and temples: Benchmarking large-scale scene reconstruction,” *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.
- [42] A. Gupta, A. Murali, D. P. Gandhi, and L. Pinto, “Robot learning in homes: Improving generalization and reducing dataset bias,” *NeurIPS*, vol. 31, 2018.
- [43] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in *2011 IEEE ICRA*, 2011, pp. 3400–3407.
- [44] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778.
- [45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, 2009, pp. 248–255.
- [46] M. Kazhdan and H. Hoppe, “Screened poisson surface reconstruction,” *ACM Trans. Graph.*, vol. 32, no. 3, July 2013.
- [47] J. Park, Q.-Y. Zhou, and V. Koltun, “Colored point cloud registration revisited,” *ICCV*, pp. 143–152, 2017.