

How Many Views Are Needed to Reconstruct an Unknown Object Using NeRF?

Sicong Pan*

Liren Jin*

Hao Hu

Marija Popović

Maren Bennewitz

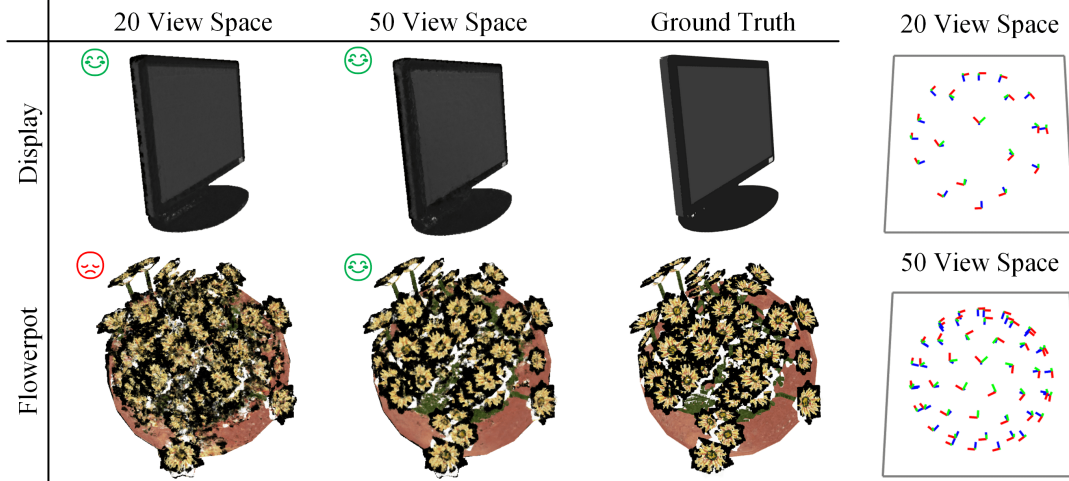


Fig. 1: An example of how object complexity affects the required number of views to reconstruct an unknown object using NeRF. The objects are trained under 20 and 50 views of hemispherical view spaces shown in the last column. The images rendered from novel test views are shown in the first two columns. As can be seen, a less colorful and geometrically simple display can be reconstructed well with 20 views, whereas a colorful and geometrically complex flowerpot requires 50 views to achieve a good result. In this work, we present an approach to predict the required number of views by a deep neural network based on the complexity of the object to be reconstructed.

Abstract—Neural Radiance Fields (NeRFs) are gaining significant interest for online active object reconstruction due to their exceptional memory efficiency and requirement for only posed RGB inputs. Previous NeRF-based view planning methods exhibit computational inefficiency since they rely on an iterative paradigm, consisting of (1) retraining the NeRF when new images arrive; and (2) planning a path to the next best view only. To address these limitations, we propose a non-iterative pipeline based on the Prediction of the Required number of Views (PRV). The key idea behind our approach is that the required number of views to reconstruct an object depends on its complexity. Therefore, we design a deep neural network, named PRVNet, to predict the required number of views, allowing us to tailor the data acquisition based on the object complexity and plan a globally shortest path. To train our PRVNet, we generate supervision labels using the ShapeNet dataset. Simulated experiments show that our PRV-based view planning method outperforms baselines, achieving good reconstruction quality while significantly reducing movement cost and planning time. We further justify the generalization ability of our approach in a real-world experiment.

I. INTRODUCTION

Object 3D reconstruction is a crucial task in robotic active vision [1], which utilizes online view planning to move the camera to maximize the information about the object to be reconstructed. Prior research mostly uses explicit 3D representations such as point clouds [2, 3], voxel grids [4, 5], and meshes [6, 7] to perform view planning. However, these methods require discretizing the scene thus leading to substantial memory consumption. Meanwhile, updating an explicit representation relies on depth-sensing modalities, *i.e.*, fusion by depth. In contrast, Neural Radiance Fields (NeRFs) [8] offer an alternative approach by implicitly modeling 3D space from a set of posed RGB images, utilizing continuous functions implemented as deep neural networks, *i.e.*, fusion by learning. Consequently, NeRFs demonstrate memory efficiency and provide high reconstruction quality. With the emergence of highly efficient training architectures like Instant-NGP [9], the integration of NeRF models into online view planning becomes feasible.

Existing NeRF-based view planning methods follow the active learning paradigm [10] to achieve good reconstruction performance, in which the most informative next-best-view (NBV), *e.g.*, the most uncertain view, is selected iteratively. The robot navigates to the NBV for new image collection until a predefined maximum number of iterations, *i.e.*, the required number of views, or a performance plateau is reached. The primary limitation lies in the fact that these methods

*These authors contributed equally to this work.

Sicong Pan and Maren Bennewitz are with the Humanoid Robots Lab, Liren Jin and Marija Popović are with the Institute of Geodesy and Geoinformation, University of Bonn, Germany. Maren Bennewitz is additionally with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany. Hao Hu is with Intel Asia-Pacific Research & Development Ltd. This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant 459376902 – AID4Crops and under Germany’s Excellence Strategy, EXC-2070 – 390732324 – PhenoRob. Corresponding author: span@uni-bonn.de

rely on greedy NBV planning based on the current NeRF state. This often necessitates retraining the NeRF when new images are collected, leading to computational inefficiency compared to faster depth fusion updates. Another drawback is that the robot only iteratively executes paths between NBVs resulting in path planning inefficiency. Motivated by the inherent inefficiencies, we develop a novel online view planning method that discards the need for iterative planning

To realize this capability, two essential components are required to plan all views at once: (1) the required number of views until the reconstruction mission can be terminated; (2) the view configuration, *i.e.*, how and where to place these views. Regarding the view configuration, we assume a hemispherical view space and simply utilize the solution to the Tammes problem [11], which finds the placement of a given number of points on a sphere to maximize the minimum distance between them. Although the theoretically optimal views should be adaptively configured based on the specific object to be reconstructed, our experimental results suggest that using the Tammes configuration is sufficiently effective. In this work, our primary focus is on discussing the problem of finding the required number of views to reconstruct a specific object.

This problem is not fully discussed in previous NeRF-based view planning literature, which relies on purely heuristic approaches or a user-defined number of views [12–15]. These methods cannot guarantee both an adequate result and a highly efficient reconstruction. In particular, complex objects usually require denser views to achieve high reconstruction quality, while less views are sufficient to reconstruct simple objects. As shown in Fig. 1, different objects have different levels of complexity, such as color, geometry, etc., and require different numbers of views to achieve a good reconstruction. Based on this observation, our novel method proposes to predict the object-specific required number of views to strike a balance between quality and efficiency in active NeRF reconstruction.

We model the relationship between the object complexity and the required number of views as a regression problem solved by a deep neural network PRVNet. We devise our PRVNet to extract features from multiple RGB images captured from initial views, thereby fostering a comprehension of the object complexity. To supervise PRVNet training, we generate a new dataset with different objects from ShapeNet [16] labeled with the required number of views. The label of the required number of views is computed by finding the minimum number of views to reach a prefixed gradient threshold of the curve representing Peak Signal-to-Noise Ratio (PSNR) performance over the number of views of a specific object. Given the number of views predicted by PRVNet, our method configures a Tammes view space and computes globally shortest paths between these views. This enables us to reduce the movement cost in contrast to iterative methods that only plan a path to the NBV.

Compared to two baselines from recent literature [12, 15], our PRV-based view planning can reconstruct an unknown object with better or comparable NeRF representation with

significantly less movement cost and planning time. The contributions of our work are threefold:

- An efficient pipeline for active NeRF reconstruction, avoiding iterative planning with time-consuming re-training and high movement cost.
- An unknown object reconstruction method based on the prediction of the required number of views, which balances between the quality and efficiency of reconstruction.
- Our PRVNet along with a dataset containing the required number of views for every object, modeling the relationship between object complexity and required number of views in NeRFs.

To support reproducibility, our implementation and dataset will be released at: <https://github.com/psc0628/NeRF-PRV>.

II. RELATED WORK

A. View Planning for Object Reconstruction

View planning methods for object reconstruction can be largely grouped into two classes: search and learning. Zeng et al. [17] summarized modern search-based methods as a generate-and-test procedure that generates a set of candidate views and tests each view by its current utility. The utility is commonly defined by intuitive concepts such as frontiers [18–20], shape analysis [6, 7, 21], occupancy with occlusion awareness [4, 22, 23], and global coverage optimization [5, 24, 25]. Deep learning-based methods treat the NBV planning problem as a classification or regression problem, which trains a network given candidate view and its potential surface coverage value as the label [2, 26–28]. Some classification networks [29, 30] output multiple views at once by learning from set-covering optimization problems. Other methods formulate a reinforcement learning framework [31–33] to learn a view planning policy from rewards in the environment.

The stopping criterion, *i.e.*, required number of views, has recently attracted the interest of researchers as it determines the stopping time and the efficiency of the reconstruction. Delmerico et al. [4] propose stopping the reconstruction when any candidate view falls below a user-defined threshold. Yervilla-Herrera et al. [34, 35] terminate the reconstruction when the number of frontier voxels is not changing (or equivalently the entropy of the frontier voxels is constant or the variation is smaller than a threshold). Pan et al. [29] utilize a deep neural network to output a set of views and use the size of the predicted view set as the required number of views. However, these studies consider an explicit map representation. Defining the stopping criterion for implicit representations remains an open problem.

B. Next-Best-View Planning in Neural Radiance Fields

Different from previous works using explicit map representations, NeRF-based methods pose challenges in quantifying the utility of the view candidates. Since the explicit geometry is not directly available from NeRFs, view selection based on surface coverage or frontiers is hard to

achieve. Emerging works study NeRF-based view planning by incorporating uncertainty quantification into NeRF representations. Pan et al. [10] learn NeRFs assuming Gaussian distribution on the radiance value and train the variance prediction by minimizing negative log-likelihood. This work adds the view candidate with the highest information gain, *i.e.*, the highest uncertainty reduction, to the existing training data. Instead of learning uncertainty prediction additionally, Lee et al. [36] exploit the entropy of the density prediction along the ray as an uncertainty measure with respect to the scene geometry. The entropy is used to guide measurement acquisition towards geometrically less precise or unexplored parts. Thanks to the recent development of fast training by Instant-NGP [9], Lin et al. [12] and Sünderhauf et al. [15] train an ensemble of NeRF models for a single scene and treat variance of the ensemble’s prediction as uncertainty quantification. Jin et al. [13] train a generalizable image-based neural rendering network together with uncertainty prediction with respect to the input data uncertainty.

The above-mentioned works focus on uncertainty quantification in NeRFs and use it for NBV planning. A key assumption is that the required number of views is defined by a user, *e.g.*, common choices are 10 and 20 total views [12, 13, 37]. Ran et al. [14] stop at 28 views and also tests 18, 38, and 58 views. Lee et al. [36] use 15 clustered views in the real world for initialization and then plan 12 NBVs. Sünderhauf et al. [15] use 5 similar views for initialization and plans NBVs up to 30 views. In contrast to these fixed constraints, we adaptively predict the number of required views based on the complexity of the object to be reconstructed, enabling us to effectively allocate the measurement budget.

III. SYSTEM OVERVIEW

Our goal is to actively reconstruct the NeRF representation of an unknown object in a tabletop scenario. This reconstruction is accomplished by utilizing a series of posed RGB images captured from various sensor views guided by a robotic arm. Fig. 2 shows the workflow of the online phase of our object reconstruction system.

The online phase begins with the robotic acquisition of three images from initial views: top, left or right, and front or back, which encompass crucial information, such as size and texture, about the object on the tabletop. The setup of initial views is confirmed in the ablation study presented in Sec. VI-B. The robot traverses three views and stops at the top view. Subsequently, we input these images into our PRVNet to predict the object-specific required number of views for reconstructing the object as detailed in Sec. V. This prediction determines the generation of a Tammes view space surrounding the object, and a global path is calculated for traversing these views as explained in Sec. IV. The robot then navigates to each view according to the global path, capturing images and saving them along with their corresponding view poses. In the offline phase, these posed images, including the three initial measurements, are used for the NeRF reconstruction.

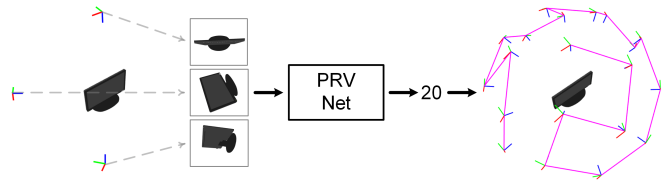


Fig. 2: An example of our online workflow given three initial views. The selected initial views (top, left, and front) are represented by red-green-blue axes. The robot takes these images and stops at the top view. We input these images into our PRVNet to obtain a predicted number of views for the reconstruction (20 in this example). Based on this, we generate the Tammes view space [11] of size 20 and the purple global path for the robot to execute.

IV. VIEW SPACE AND PATH PLANNING

We assume a hemispherical view space on the tabletop with views pointing to the center of the hemisphere, as often considered in active object reconstruction approaches [10, 13–15, 29, 36]. The position of each view is defined by Tammes problem [11] that solves the task of placing a given number of points on a sphere to maximize the minimum distance between them.

Our global path planning method solves the problem of connecting all views in the Tammes view space. We generate the optimal global path by solving the shortest Hamiltonian path problem on a graph, which is similar to the traveling salesman problem (TSP) but without returning to the starting node. As Gurobi Optimizer [38] efficiently resolves TSP (less than 100 nodes) within seconds, we introduce a virtual starting node to convert the Hamiltonian path problem into a TSP scenario and efficiently obtain the final robot global path. An illustration of the global path is given in Fig. 2. The view-to-view local path follows the concept of avoiding the object as an obstacle on the tabletop as fully defined in [30].

V. PREDICTING REQUIRED NUMBER OF VIEWS IN NeRF

This section presents our novel PRVNet, which is designed to adaptively determine the required number of views for a specific object. For network training, we generate a dataset consisting of individual objects and their required number of views. Given initial measurements of an object as input, PRVNet is trained under the supervision of the corresponding view-number label.

A. Object-Specific Required Number of Views

Our proposed approach is based on the key insight that as the object complexity increases, a larger number of views is necessary to obtain a good NeRF representation. To quantitatively study this relationship, we plot the PSNR value for a specific object over the number of views $v \in \mathbb{N}^+$, ranging from 3 to 50 at intervals of 2. Fig. 3 illustrates the plots for two example objects. While the PSNR values may fluctuate due to the inherent training randomness in CUDA [9], we observe that with a higher count of views, the rate of PSNR growth diminishes to zero for a specific object. We exploit these convergence trends to assess the complexity levels of different objects.

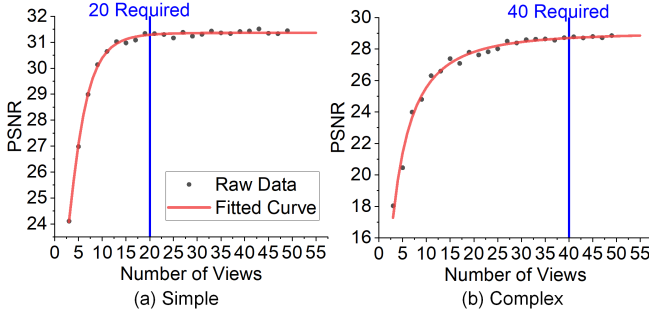


Fig. 3: An example of quantitative analysis of the required number of views on different object complexity: (a) a simple object, (b) a complex object. Each black point is a pair of (v, PSNR) , which means a NeRF trained under a view space of size v , and images from 100 test Tammes views are rendered to report an average PSNR value. The red curve C_o is fitted to these data points to determine the v^* based on its gradient. The blue lines suggest that for a simple object, we achieve a satisfactory result with only 20 views, whereas a complex object necessitates 40 views.

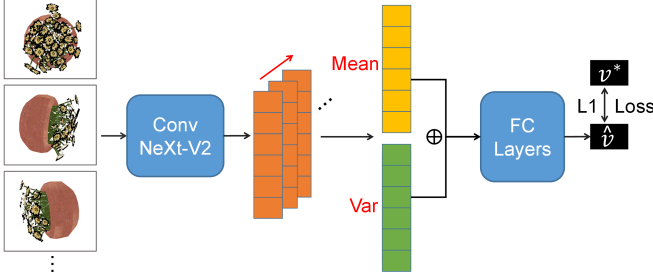


Fig. 4: PRVNet architecture: We use ConvNeXt-V2 [39] as the backbone to extract features from each image. The red arrow indicates the calculation of mean and variance across the batch dimension. \oplus represents the concatenation operation. L1 loss is employed for network training.

B. Labeling of Required Number of Views

To ensure the monotonic increase of the growth rate used for labeling, we use a curve-fitting approach to mitigate the impact of fluctuations. Given that the growth rate typically follows a skewed distribution (decreasing as the number of views increases), a log-normal distribution is often assumed [40]. Consequently, the raw data can be fitted to a cumulative distribution function, denoted as C_o , as depicted in Fig. 3. When the gradient of the growth rate falls below a certain small threshold α , we deduce that a required number of views $v^* \in \mathbb{N}^+$ is sufficient for an object o to achieve a satisfactory NeRF representation:

$$v^* = \operatorname{argmin}(v), \text{ s.t. } C_o(v+1) - C_o(v) < \alpha. \quad (1)$$

Once the required number of views label v^* is computed for an object o , we generate the supervision pair (I_o, v^*) for the network training, where I_o is the list of images taken from several initial views around object o .

C. Learn to Predict the Required Number of Views

Our PRVNet is a regression network that takes several images I_o as input. To enable the PRVNet to process multiple image inputs and learn multi-view information, we devised

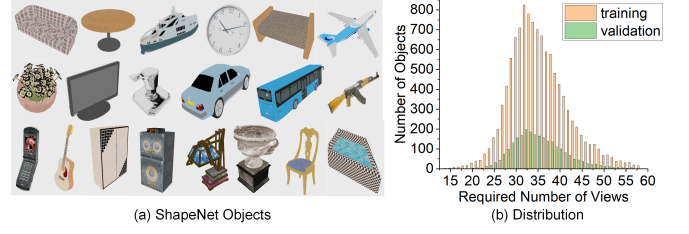


Fig. 5: 3D model and required number of views datasets: (a) textured examples of the top 20 classes in ShapeNet [16]; (b) training and validation set distributions over required number of views.

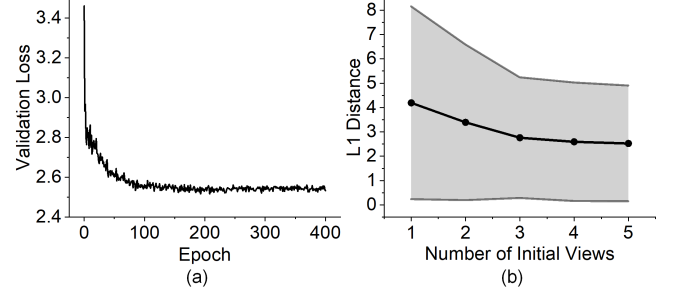


Fig. 6: PRVNet training: (a) validation loss over epochs, (b) L1 distance with its standard deviation over the number of initial views (the black curve with the gray error bar).

a network architecture illustrated in Fig. 4. The output of the last Sigmoid activation is converted to the range of required numbers of views in our dataset by linear mapping and constant offset to acquire final prediction $\hat{v} \in \mathbb{R}^+$. We use L1 loss to enforce our PRVNet to predict a value close to the ground truth label. During deployment, \hat{v} is rounded to an integer for configuring Tammes view space.

VI. EXPERIMENTS

Our experiments are designed to support the claim that our method can achieve fast online data collection for high-quality NeRF reconstruction of an unknown object by predicting the required number of views.

A. Dataset Generation

Object 3D Model Dataset. The capabilities of our method depend on a well-trained PRVNet. We generate the required number of views dataset on the ShapeNet 3D mesh model dataset [16] that contains different classes of objects. Given the imbalanced distribution of objects across various classes within the ShapeNet, we consider a maximum of 1,200 objects per class for the top 20 classes as shown in Fig. 5(a). On the other hand, texture information is important to object complexity. We therefore only consider 3D models with textures and use a sampling method [41] to ensure the visual result is the same as the original mesh.

Required Number of Views Dataset. We perform virtual imaging of resolution 1280×720 px on object 3D models from different views in a simulation environment. Considering a real-world tabletop environment, we set the radius of view spaces to 0.3 m. Since object size can also be interpreted as part of the object complexity, we randomize

Method	Required Views	PSNR Difference ↓	SSIM Difference ↓	Movement Difference (m) ↓
GT Label	34.94 ± 5.70	0	0	0
Mode	32	0.1641 ± 0.1551	0.002163 ± 0.002370	0.3070 ± 0.3117
Median	34	0.1511 ± 0.1630	0.002248 ± 0.002397	0.2896 ± 0.2624
Mean	35	0.1624 ± 0.1581	0.002086 ± 0.002278	0.2913 ± 0.2458
PRVNet (Proposed)	35.77 ± 5.24	0.1390 ± 0.1588	0.001817 ± 0.001928	$*0.1988 \pm 0.2032$

TABLE I: Comparison to statistic methods. We report five different methods to compute the required number of views: Ground Truth Label (GT Label), Mode 32, Median 34, Mean 35 shown in Fig. 5(b), as well as our PRVNet. PSNR and SSIM are computed from 100 novel views. The Movement/PSNR/SSIM Difference stands for the absolute difference from the GT Label. Each value reports the average mean and standard deviation on 250 random objects from the validation set. The star indicates significant results against statistic methods according to the paired t -test with a p -value of 0.05. As can be seen, our PRVNet output aligns most closely with the GT Label, indicating a good prediction of the required number of views based on object complexity.

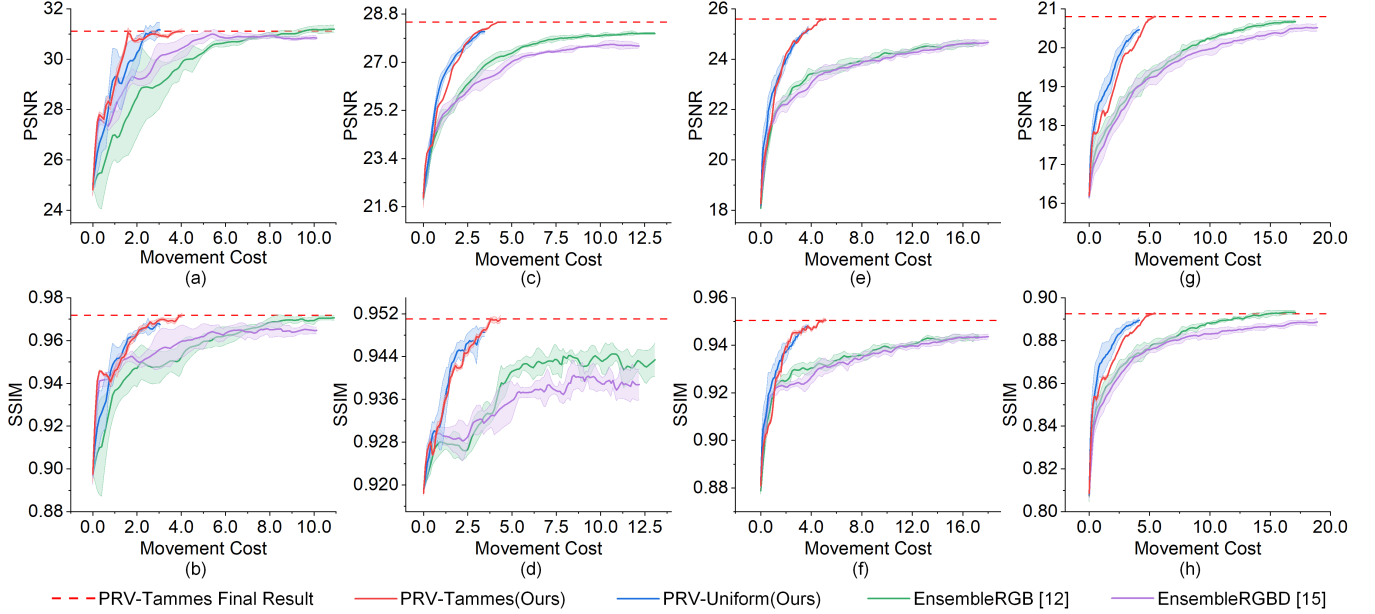


Fig. 7: Comparison of view planning results. We report four objects with different complexity, *i.e.*, required number of views in ground truth labels varying from small to large: (a), (b) Simple 20; (c), (d) Relatively Simple 30; (e), (f) Relatively Complex 40; (g), (h) Complex 50. PSNR and SSIM are computed from 100 novel views. Each row shows PSNR and SSIM, respectively, over the movement cost, which is computed by linear interpolation with the interval 0.1 because only iterative data are available. Each planner is tested 5 times and its average mean with standard deviation (the error bar) is reported. As can be seen, (1) our PRV-based methods achieve superior or comparable PSNR/SSIM while requiring less movement cost than other baselines, particularly for more complex objects; (2) our Tammes view configuration achieves better reconstruction with smaller standard deviations than uniformly sampled configurations.

the object size from 0.07 m to 0.12 m as data augmentation. After obtaining these ground truth images from 3-50 view spaces, we train a NeRF for each object and required number of view pairs to get the PSNR data plots for fitting C_o as discussed in Sec. V-A. The training and rendering of these NeRFs are implemented using Instant-NGP [9] with a training step of 2,500 on a cluster of 8 NVIDIA A100 Tensor Core GPUs. The Orthogonal Distance Regression method in OriginPro [42] is used for curve fitting. In total, we label 13,789 objects with the required number of views under $\alpha = 0.02$. We employ an 8/2 ratio to randomly partition our dataset into training and validation set as shown in Fig. 5(b).

B. Network Training

Implementation Details and Parameters. We use the tiny model for ConvNeXt-V2 and set the output feature layer as 1,000. The fully connected (FC) layers are then sized as [1,000, 500, 250, 100, 1]. The output of PRVNet is remapped to [13, 58] as the range of views in our dataset shown in

Fig. 5(b). The batch size is set to 64, the base learning rate is set to 0.00015, and the weight decay is set to 0.05. The pre-trained weights of ConvNeXt-V2 on ImageNet-1K are used in our PRVNet for better feature extraction. The size of I_o is set to the number of initial views (top, left, right, front, and back) used for training. We train our PRVNet for 400 epochs on 8 A100 GPUs. The validation loss over epochs is shown in Fig. 6(a). We save the network with the smallest L1 loss on the validation set as the final result.

Ablation Study on Initial Views. Fig. 6(b) reports the results from different numbers of initial views on the validation set. Although the best results could be achieved with five views, the setup of three views is stable enough to use. We finally chose $|I_o| = 3$ as input for the network to improve the reconstruction speed.

Comparison to Statistic Methods. We perform a comparison with the basic statistics, *i.e.*, the number of views for each object using a specific statistic, as shown in Table I. From the results, we confirm that the PSNR, SSIM (Struc-

Method	PSNR \uparrow	SSIM \uparrow	Movement Cost (m) \downarrow	Planning Time (s) \downarrow
EnsembleRGB [12]	26.96 ± 2.86	0.9419 ± 0.0879	12.719 ± 2.510	2536.5 ± 500.1
EnsembleRGBD [15]	27.09 ± 2.17	0.9526 ± 0.0239	12.458 ± 2.440	2600.3 ± 521.6
PRV-Uniform (Ours)	27.49 ± 2.25	0.9562 ± 0.0230	3.336 ± 0.316	0.687 ± 0.039
PRV-Tammes (Ours)	27.84 ± 2.25	0.9577 ± 0.0229	4.589 ± 0.372	0.605 ± 0.003

TABLE II: Comparison of final reconstruction results. We report the metrics after all images are collected. PSNR and SSIM are computed from 100 novel views. The movement cost and planning time are total sum values during the online reconstruction. Each value is reported as the averaged mean and standard deviation on 50 random objects from the validation set. Two ensemble baselines [12, 15] are assumed to be fully paralleled and the reported time is divided by the number of ensembles. Note that the standard deviations primarily arise from the complexity of different objects. As can be seen, our PRV-Tammes method is highly efficient in terms of movement cost and planning time along with better quality.

tural Similarity Index) [8], and movement cost differences (with respect to the ground truth results) of the PRVNet are smaller than the statistical methods. This means that the PRVNet effectively predicts the appropriate required number of views for objects of varying complexity, *i.e.*, giving an object-specific prediction.

C. Evaluation of View Planning

Baselines and Metrics. We compare our PRV-based view planning method (PRV-Tammes) with two uncertainty-based NBV methods (EnsembleRGB [12] and EnsembleRGBD [15]). We set a planning view space of size 540×45 for the baselines. The resolution of ensembles is set to 90×45 to have a rapid uncertainty rendering. In addition, we also perform an ablation study on our Tammes configuration by replacing it with uniformly sampled views (PRV-Uniform), *i.e.*, random sampling views from the planning view space with the number of the PRVNet output. The global path planning is also used for PRV-Uniform. We evaluate the methods on an i7-12700H CPU and an Nvidia RTX3060 Laptop GPU to represent deployment scenarios. We use PSNR and SSIM to evaluate the quality of NeRF representations. The movement cost, and planning time are used to evaluate the efficiency of view planning.

Setup and Results. For a fair comparison, the same three initial views (top, left, and front) are configured for each method, and the number of views is set as the same as the output from the PRVNet. Note that, unlike the previous comparison of statistics, the images of the initial views are also included in the NeRF training. Two sources of randomness influence the planning results: (i) the Instant-NGP training process; (ii) the planning methods. We thus perform multiple trials for each planner on four objects from simple to complex to explore these randomnesses in Fig. 7, as well as the final reconstruction results on more object cases in Table II. From the results, we confirm that: (1) Our PRV-Tammes method achieves higher or similar PSNR/SSIM quality within less movement cost than other baselines, especially for the more complex objects. (2) Our PRV-based methods require very little planning time, *i.e.*, inferring with the PRVNet once, compared to iterative baselines, where retraining NeRF is required between planning steps. (3) Our PRV-Uniform method also achieves high reconstruction efficiency but lower final PSNR/SSIM and is less stable (larger standard deviation in Fig. 7) than our Tammes configuration.

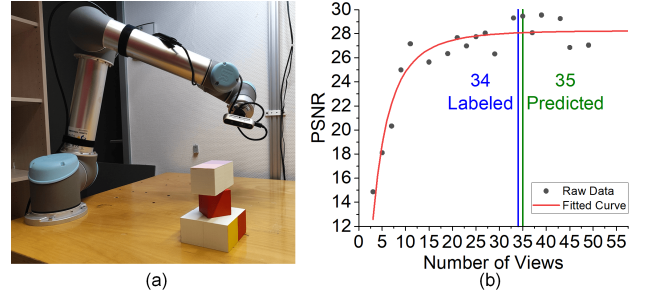


Fig. 8: Real-world reconstruction: (a) robot environment and the object, (b) generalization test. The required number of views labeled is computed by the fitted PSNR curve as 34 (blue line). Our PRVNet takes three initial images and outputs 35 (green line). The small difference in the required view number indicates that our PRVNet has a reasonable sim-to-real performance.

D. Real-World Reconstruction

Setup. We validate our approach in a real-world environment using a UR5 robot arm with an Intel Realsense D435 camera mounted on its end-effector (only the RGB optical camera is activated). ROS [43] and MoveIt [44] are used for robotic motion planning.

Generalization and Reconstruction Test. We collect real-world images from different Tammes view spaces of an object to compute the label of the required number of views introduced in Sec. V-B. The experimental environment and data are shown in Fig. 8. The online data collection process and final reconstruction results are presented in the accompanying video¹. From the results, we confirm that: (1) Our PRVNet can generalize to real-world environments, and (2) our PRV-based view planning achieves fast online image collection and good NeRF reconstruction quality.

VII. CONCLUSION

In this paper, we present a novel non-iterative pipeline for active NeRF reconstruction using the prediction of the required number of views and the Tammes configuration for view pose generation. We propose PRVNet trained on our new dataset consisting of objects of different complexities to predict the required number of views. We leverage the network output to plan a globally connected path representing the minimum travel distance. Our experiments show that our view planning using the PRVNet prediction achieves a higher efficiency in terms of movement cost and competitive quality in reconstruction compared to state-of-the-art baselines.

¹<https://youtu.be/LoQGOR3S1Fw>

REFERENCES

- [1] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.
- [2] R. Zeng, W. Zhao, and Y.-J. Liu, "Pc-nbv: A point cloud based deep network for efficient next best view planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, 2020, pp. 7050–7057.
- [3] Border, Rowan and Gammell, Jonathan D, "The surface edge explorer (see): A measurement-direct approach to next best view planning," *arXiv preprint arXiv:2207.13684*, 2022.
- [4] 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.
- [5] S. Pan and H. Wei, "A global max-flow-based multi-resolution next-best-view method for reconstruction of 3d unknown objects," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 714–721, 2022.
- [6] S. Wu, W. Sun, P. Long, H. Huang, D. Cohen-Or, M. Gong, O. Deussen, and B. Chen, "Quality-driven poisson-guided autoscanning," *ACM Transactions on Graphics*, vol. 33, no. 6, 2014.
- [7] I. D. Lee, J. H. Seo, Y. M. Kim, J. Choi, S. Han, and B. Yoo, "Automatic pose generation for robotic 3-d scanning of mechanical parts," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1219–1238, 2020.
- [8] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [9] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.
- [10] X. Pan, Z. Lai, S. Song, and G. Huang, "ActiveNeRF: Learning Where to See with Uncertainty Estimation," in *Proc. of the Europ. Conf. on Computer Vision*, 2022.
- [11] X. Lai, D. Yue, J.-K. Hao, F. Glover, and Z. Lü, "Iterated dynamic neighborhood search for packing equal circles on a sphere," *Computers & Operations Research*, vol. 151, p. 106121, 2023.
- [12] K. Lin and B. Yi, "Active view planning for radiance fields," in *Robotics Science and Systems Workshop*, 2022.
- [13] L. Jin, X. Chen, J. Rückin, and M. Popović, "Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, to be published, 2023.
- [14] Y. Ran, J. Zeng, S. He, J. Chen, L. Li, Y. Chen, G. Lee, and Q. Ye, "Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1125–1132, 2023.
- [15] N. Sinderhauf, J. Abou-Chakra, and D. Miller, "Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9370–9376.
- [16] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.
- [17] 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.
- [18] R. Border, J. D. Gammell, and P. Newman, "Surface edge explorer (see): Planning next best views directly from 3d observations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6116–6123.
- [19] R. Border and J. D. Gammell, "Proactive estimation of occlusions and scene coverage for planning next best views in an unstructured representation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4219–4226.
- [20] T. Zaenker, C. Lehnert, C. McCool, and M. Bennewitz, "Combining local and global viewpoint planning for fruit coverage," in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–7.
- [21] R. Menon, T. Zaenker, and M. Bennewitz, "Nbv-sc: Next best view planning based on shape completion for fruit mapping and reconstruction," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, to be published, 2023.
- [22] J. Daudelin and M. Campbell, "An adaptable, probabilistic, next-best view algorithm for reconstruction of unknown 3-d objects," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1540–1547, 2017.
- [23] T. Zaenker, C. Smitt, C. McCool, and M. Bennewitz, "Viewpoint planning for fruit size and position estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3271–3277.
- [24] S. Pan and H. Wei, "A global generalized maximum coverage-based solution to the non-model-based view planning problem for object reconstruction," *Computer Vision and Image Understanding*, vol. 226, p. 103585, 2023.
- [25] T. Zaenker, J. Rückin, R. Menon, M. Popović, and M. Bennewitz, "Graph-based view motion planning for fruit detection," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, to be published, 2023.
- [26] 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.
- [27] J. I. Vasquez-Gomez, D. Troncoso, I. Becerra, E. Sucar, and R. Murrieta-Cid, "Next-best-view regression using a 3d convolutional neural network," *Machine Vision and Applications*, vol. 32, no. 2, pp. 1–14, 2021.
- [28] Y. Han, I. H. Zhan, W. Zhao, and Y.-J. Liu, "A double branch next-best-view network and novel robot system for active object reconstruction," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7306–7312.
- [29] S. Pan, H. Hu, and H. Wei, "Scvp: Learning one-shot view planning via set covering for unknown object reconstruction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1463–1470, 2022.
- [30] S. Pan, H. Hu, H. Wei, N. Dengler, T. Zaenker, and M. Bennewitz, "One-shot view planning for fast and complete unknown object reconstruction," *arXiv preprint arXiv:2304.00910*, 2023.
- [31] D. Peralta, J. Casimiro, A. M. Nilles, J. A. Aguilar, R. Atienza, and R. Cajote, "Next-best view policy for 3d reconstruction," in *2020 European Conference on Computer Vision*. Glasgow, UK: Springer, 2020, pp. 558–573.
- [32] X. Zeng, T. Zaenker, and M. Bennewitz, "Deep reinforcement learning for next-best-view planning in agricultural applications," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2323–2329.
- [33] N. Dengler, S. Pan, V. Kalagaturu, R. Menon, M. Dawood, and M. Bennewitz, "Viewpoint push planning for mapping of unknown confined spaces," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, to be published, 2023.
- [34] H. Yervilla-Herrera, J. I. Vasquez-Gomez, R. Murrieta-Cid, I. Becerra, and L. E. Sucar, "Optimal motion planning and stopping test for 3-d object reconstruction," *Intelligent Service Robotics*, vol. 12, pp. 103–123, 2019.
- [35] H. Yervilla-Herrera, I. Becerra, R. Murrieta-Cid, L. E. Sucar, and E. F. Morales, "Bayesian probabilistic stopping test and asymptotic shortest time trajectories for object reconstruction with a mobile manipulator robot," *Journal of Intelligent & Robotic Systems*, vol. 105, no. 4, p. 82, 2022.
- [36] S. Lee, L. Chen, J. Wang, A. Liniger, S. Kumar, and F. Yu, "Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 070–12 077, 2022.
- [37] H. Zhan, J. Zheng, Y. Xu, I. Reid, and H. Rezafofighi, "Activermap: Radiance field for active mapping and planning," *arXiv preprint arXiv:2211.12656*, 2022.
- [38] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021.
- [39] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 133–16 142.
- [40] E. L. Crow and K. Shimizu, *Lognormal distributions*. Marcel Dekker New York, 1987.
- [41] D. Lazzarotto and T. Ebrahimi, "Sampling color and geometry point clouds from shapenet dataset," *arXiv preprint arXiv:2201.06935*, 2022.
- [42] O. Corporation, "Originpro," 2021.
- [43] A. Koubâa et al., *Robot Operating System (ROS)*. Springer, 2017, vol. 1.
- [44] S. Chitta, "Moveit!: an introduction," *Robot Operating System (ROS) The Complete Reference (Volume 1)*, pp. 3–27, 2016.