2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Perceiving Unseen 3D Objects by Poking the Objects

Linghao Chen Yunzhou Song Hujun Bao Xiaowei Zhou
State Key Lab of CAD&CG, Zhejiang University

***Abstract*— We present a novel approach to interactive 3D object perception for robots. Unlike previous perception algorithms that rely on known object models or a large amount of annotated training data, we propose a poking-based approach that automatically discovers and reconstructs 3D objects. The poking process not only enables the robot to discover unseen 3D objects but also produces multi-view observations for 3D reconstruction of the objects. The reconstructed objects are then memorized by neural networks with regular supervised learning and can be recognized in new test images. The experiments on real-world data show that our approach could unsupervisedly discover and reconstruct unseen 3D objects with high quality, and facilitate real-world applications such as robotic grasping. The code and supplementary materials are available at the project page: `https://zju3dv.github.io/poking_perception/`.**

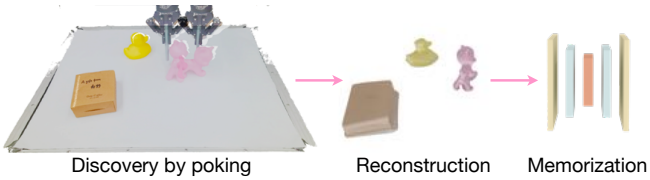

Fig. 1. **The proposed system for 3D objects perception.** The poking process not only enables the system to discover unseen 3D objects but also provides multi-view observations for object reconstruction. Based on the reconstructed object models, the objects are memorized by neural networks for recognizing them on new test images.

I. INTRODUCTION

3D object perception plays a crucial role in computer vision and robotics, with numerous real-world applications, such as grasping, manipulation, and scene understanding. Most existing methods for object perception either rely on known object models or a large number of annotated data for training. Since these approaches are costly and limited to a single object instance or a few categories presented in the training data, they are hardly applicable in real-world scenarios, where many unseen objects may exist. Imagine that a robot enters a new environment containing some objects it has never seen before, how would it perceive the 3D objects for subsequent operations?

Typically, humans understand their surroundings through interactive perception. By interacting with objects in the scene, such as pushing, grasping, or poking, they can identify the objects and build their 3D representations, which finally serve as a knowledge base to recognize them once presented again. In this work, we present a novel system that imitates this human behavior. As shown in Fig. 1, 3D object discovery is achieved by poking, which enables the system to handle unseen 3D objects regardless of their shapes, appearances, categories, and poses. The poking process generates multi-view observations for the 3D objects by motion, which are used to reconstruct 3D object models. The reconstructed models are then memorized through neural networks, which are used for object recognition on new test images.

Specifically, given a scene with several unseen objects, we first generate object proposals through point cloud clustering based on geometric assumptions, which are then examined by poking with a robot arm. The poking process prunes immovable object proposals and generates multi-view observations of 3D objects. We then use implicit neural representation learning to reconstruct the objects based on these multi-view observations, which optimizes geometry, appearance, and poses simultaneously to yield high-quality object models. Finally, the reconstructed models are memorized through training a detector or object pose estimator with images rendered from the models. The memorization process allows us to recognize and perceive these objects with only one forward pass on a new test image, enabling various downstream tasks in real-world applications, such as robotic grasping, manipulation, and scene understanding.

We evaluate our system through experiments in real-world scenes. The results show that our method can effectively discover unseen 3D objects and reconstruct them with high quality in terms of geometry, appearance, and poses. Additionally, the memorized object models enable precise detection and pose estimation of the objects on new test images.

Corresponding author: Xiaowei Zhou.

II. RELATED WORK

Interactive perception. Currently, most 3D perception tasks are passive, such as object detection [1]–[3], object pose estimation [4, 5], object reconstruction [6, 7], etc. These methods either rely on known object models or large amounts of annotated data for training, which limits their applicability in the real world. In contrast, several recent works in robotics propose to learn from interaction with the environment [8]. [9] learns to map poking to object motion by random poking and recording the change in the visual state of the world. [10, 11] learn the object-centric representation to build the mapping between physics actions and visual observations. DensePhysNet [12] and DSR-Net [13] are most relevant to ours. DensePhysNet [12] proposes to perform

979-8-3503-2365-8/23/$31.00 ©2023 IEEE

a few dynamic interactions with objects to learn a dense object representation, and DSR-Net [13] proposes to use interactive perception to discover, track, and reconstruct objects simultaneously. However, relying on a set of predefined object categories or models for training limits their abilities in generalizing to unseen objects. Recently, several works in computer vision propose to discover and perceive 3D objects by motion. [14] and [15] propose unsupervised training approaches to decompose the dynamic scene into the background and several moving objects using motion cues. However, all of them struggle with real-world scenes due to the large gap between synthetic and real-world data in terms of the visual complexity and diversity of object geometries and appearances.

Robotic grasping. Traditionally, the simulator Graspit! [16] generates a grasp through several analytical methods given the object model. Recent works [17]–[24] propose learning-based approaches to learn grasping from a large amount of labeled data. Given a depth image as input, they predict the grasp in an end-to-end manner to avoid the difficult problem of reconstructing the high-quality object model. However, the lack of reasoning of object properties, such as geometry and semantics, limits their applicability in downstream tasks. To tackle this problem, some methods propose to perform object reconstruction and grasping simultaneously. [25] uses the structure of the reconstruction network to classify the successful rate of grasping and use it as the objective function for continuous grasp optimization. The reconstruction could be used to further avoid undesired contact during grasping.

3D reconstruction. Traditionally, the seminal work KinectFusion [26] proposes to first estimate the sensor pose using a coarse-to-fine ICP algorithm and then perform TSDF fusion [27] to obtain the object geometry. MaskFusion [28] and MidFusion [29] perform instance segmentation before tracking and fusion to tackle the problem of reconstructing multiple moving objects. Recently, implicit neural representation learning has been widely used in the 3D reconstruction. NeRF [30] is a pioneer work that proposes to use an MLP to predict color and density for each 3D point, which is learned by inverse volume rendering. VolSDF [31] and NeuS [32] propose to predict Signed Distance Function (SDF) instead of density to increase reconstruction quality. [33, 34] propose to represent the scene with several neural radiance fields, each representing a foreground object or the background, to enable scene decomposition and editing. BaRF [35], NeRF−− [36], and STaR [37] propose to jointly optimize the parameters of neural radiance fields and the relative poses between the object and the camera to reduce reliance on accurate camera/object poses in real-world applications.

III. METHOD

Given a 3D scene with several objects, our goal is to enable a robot to perceive the existence and poses/geometries of the objects which are never seen before. Our pipeline consists of three stages: we first discover the 3D objects by poking (Sec. III-A), then reconstruct the 3D objects (Sec. III-B), and finally memorize them for recognition on new test images (Sec. III-C).

A. Object discovery by poking

We start by describing the poking process that discovers the objects in the scene and provides input to the reconstruction module.

The poking process consists of two stages. The first stage generates object proposals in the scene, which are then poked and examined in the second stage.

Since there exist infinite poking trajectories without any prior of object locations, we propose to first generate some object proposals and then examine them to reduce the poking search space which is analogous to the Region Proposal Network (RPN) in object detection [2, 38]. Specifically, assuming that objects are always lying on a plane, we first perform plane segmentation and then cluster the point clouds above the plane to obtain the object proposals. The object proposals are then examined by poking and the ones which cannot be moved will be treated as negative proposals and pruned.

After generating object proposals, a robot arm pokes each object and the process is recorded using an RGB-D camera. The design of poking trajectory only needs to ensure the objects to be viewed from an adequate number of viewpoints and avoids occlusions from the robot arm, which is achieved by performing multiple iterations of poking in a clockwise direction. The details of the heuristic-based poking policy are described in Algorithm 1 of the supplementary material.
Discussion. The utilization of learning-based grasp detection, where a neural network is employed for grasp detection followed by object grasping and scanning, is an intuitive alternative for object discovery in robotics. However, this approach is plagued by several limitations: 1) Learning-based grasp detection is limited to the training domain and may fail on unseen objects and even damage the fragile objects; 2) Some objects may be too large to be grasped; 3) Grasping may occlude the object and make the complete reconstruction difficult. In contrast, poking is neither limited by object categories or sizes nor does it introduce severe occlusion. Another alternative is to obtain multi-view observations by moving a camera instead of moving the objects in the scene. However, this approach has difficulty in segmenting objects from the scenes with complex backgrounds or when the objects are close to each other. Moreover, it cannot eliminate the occlusion between objects. In contrast, our method effectively reduces occlusion, prunes the negative object proposals and ensures the correct number of objects thanks to the poking process.

B. Object reconstruction

1) Decomposed neural radiance fields: Given the RGB-D video recorded in Sec. III-A, we devise an implicit neural representation-based approach to reconstruct the objects.

NeRF [30] represents a scene with a neural radiance field. Taking as input a 3D point $\mathbf{x}$ and a viewing direction $\mathbf{d}$, a

multilayer perceptron (MLP) is used to produce the density σ and color c of the point $\mathbf{x}$. Then the pixel color along a ray is computed using volume rendering:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i, \tag{1}$$

where N is the number of 3D points along the ray $\mathbf{r}$, $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ is a ray with origin $\mathbf{o}$ and direction $\mathbf{d}$, $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$, $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ is the accumulated transmittance along the ray, and $\delta_i = t_{i+1} - t_i$ is the distance between neighboring samples along the ray.

As a single neural radiance field could only represent one static scene, we propose to represent our dynamic scene with a decomposed neural radiance field, in which each sub-field represents a rigid part in the scene (the background or an object) similar to [34, 37].

Meanwhile, since there is no surface constraint in the NeRF representation, we follow VolSDF [31] to represent the object neural radiance field as SDF and color for high-quality reconstruction.

Denoting F_Θ^b as the background NeRF, F_Θ^k as the k-th object VolSDF ($k = 1, \cdots, K$), and $\xi_t^k \in \mathfrak{se}(3)$ as the pose of the k-th object at frame t, for a point $\mathbf{x}$ with viewing direction $\mathbf{d}$, the color and density are computed as follows:

$$\mathbf{c}(\mathbf{x})^b, \sigma(\mathbf{x})^b = F_\Theta^b(\mathbf{x}, \mathbf{d}), \tag{2}$$

$$\mathbf{c}(\mathbf{x})^k, d(\mathbf{x})^k = F_\Theta^k(\mathbf{x_o}, \mathbf{d}), \tag{3}$$

$$\sigma(\mathbf{x})^k = \begin{cases} \frac{1}{\beta}\left(1 - \frac{1}{2}\exp\left(\frac{d(\mathbf{x})^k}{\beta}\right)\right) & \text{if } d(\mathbf{x})^k < 0 \\ \frac{1}{2\beta}\exp\left(-\frac{d(\mathbf{x})^k}{\beta}\right) & \text{if } d(\mathbf{x})^k \geq 0, \end{cases} \tag{4}$$

where $d(\mathbf{x})^k$ is the signed distance of point $\mathbf{x}$, $\mathbf{x_o} = (\xi_t^k)^{-1}\mathbf{x}$ is the transformed point from the world coordinate to the object coordinate, and β is a learnable parameter.

Then, the pixel color $\hat{C}(\mathbf{r})$ and depth $\hat{D}(\mathbf{r})$ can be computed as follows:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (\alpha_i^b \mathbf{c}_i^b + \sum_{k=1}^{K} \alpha_i^k \mathbf{c}_i^k), \tag{5}$$

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{d}_i, \tag{6}$$

where K is the number of neural radiance fields, $\bar{\sigma_i} = \sigma_i^b + \sum_{k=1}^{K} \sigma_i^k$ is the composed density of all the neural radiance fields for point $\mathbf{x}_i$, $\alpha_i = 1 - \exp(-\bar{\sigma_i}\delta_i)$, $\alpha_i^k = \frac{\sigma_i^k}{\bar{\sigma_i}}\alpha_i$, $\alpha_i^b = \frac{\sigma_i^b}{\bar{\sigma_i}}\alpha_i$, and $\mathbf{d}_i$ is the depth of the point $\mathbf{x}_i$.

2) Optimizing neural radiance fields and object motion: During optimization, we jointly optimize the parameters of the neural radiance fields F_Θ^b and F_Θ^k and the object poses ξ_t^k.

Given the rendered pixel color $\hat{C}(\mathbf{r})$ and depth $\hat{D}(\mathbf{r})$, we compute the color loss and depth loss as follows:

$$\mathcal{L}_\mathrm{c} = \left\| \hat{C}(\mathbf{r}) - C(\mathbf{r}) \right\|, \tag{7}$$

$$\mathcal{L}_\mathrm{d} = \left\| \hat{D}(\mathbf{r}) - D(\mathbf{r}) \right\|, \tag{8}$$

where $\|\cdot\|$ is the 1-norm, $C(\mathbf{r})$ and $D(\mathbf{r})$ are the ground-truth color and depth of ray $\mathbf{r}$.

Moreover, we apply the Eikonal loss [39] to encourage d to approximate a signed distance function as suggested in [31].

$$\mathcal{L}_\mathrm{sdf} = \mathbb{E}_{\boldsymbol{z}}(\|\nabla d(\boldsymbol{z})\| - 1)^2, \tag{9}$$

Since the object and the background are in contact, we find it hard to decompose them especially with textureless background due to its motion ambiguity. Inspired by [40], we propose the following sparsity loss to solve this problem:

$$\mathcal{L}_\mathrm{sp} = w_{sp} \left| 1 - \exp(-\sigma_i) \right|, \tag{10}$$

where $w_{sp} = \exp(-\mathbf{w} \cdot \max(z_m - z_i, 0))$ is the loss weight of the sparsity loss, σ_i and z_i are the density and depth of a point x_i on a ray $\mathbf{r}$, $z_m = \max_t\{D_\mathbf{r}^t\}$ is the maximum depth of the ray $\mathbf{r}$ across all the frames, and $\mathbf{w}$ is a weight decay parameter.

The sparsity loss encourages the density of the object VolSDF to be small, and w_{sp} assigns different weights for points with different distances to the background surface. Intuitively, the points on and farther than the background surface are assigned a large loss weight, while the points nearer than the background surface are assigned a small one. This design eliminates the density of objects in unobserved and ambiguous spaces and reduces the effect of the sparsity loss on the spaces nearer than the background surface.

Combining the above terms, the total loss function is

$$\mathcal{L}(\Theta_b, \Theta_o, \xi_o) = w_1\mathcal{L}_\mathrm{c} + w_2\mathcal{L}_\mathrm{d} + w_3\mathcal{L}_\mathrm{sdf} + w_4\mathcal{L}_\mathrm{sp}. \tag{11}$$

Once the object neural radiance field is optimized, the object mesh is extracted with the marching cubes [41] operation, and the vertex colors are obtained by averaging the radiance at the vertex positions under all view directions in the input video. The segmentation mask could be rendered by setting the radiance of the object VolSDF to 1 and the density of the background NeRF to 0. This representation allows the network to optimize the segmentation mask implicitly and leads to a more accurate segmentation mask as demonstrated in Sec. IV-B.

Sampling strategy. Since the region of the objects is relatively small compared to the entire image, we design a foreground sampling strategy for faster convergence. Representing N_r as the number of pixels to sample over an image, we propose to sample $N/2$ pixels within the object mask and the rest $N/2$ pixels over the entire image.

Moreover, we find it difficult to decompose the robot arm and objects since they are in contact during the poking process. To restrict the impact of the robot, we propose not to sample pixels within the robot mask, which is obtained by rendering the robot arm model with its pose in each frame.

Training strategy. To avoid local optima when jointly optimizing the object poses and the neural radiance fields, we initialize the object masks and poses and propose a stage-wise training strategy. The object mask is initialized

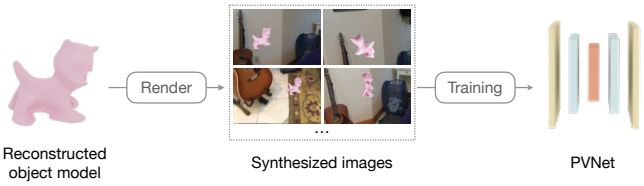

Fig. 2. **The training pipeline for PVNet based on the reconstructed object model.** The background of the synthesized images are randomly chosen from the ScanNet dataset.

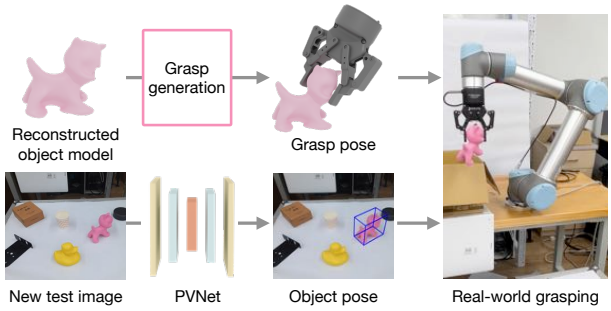

Fig. 3. **Real-world grasping pipeline based on the reconstructed object model.** Graspit! is used to generate a grasp pose given the reconstructed object model in the object coordinates and the trained PVNet is used to estimate the object pose on the new test image.

as the set of pixels whose optical flow norm is larger than a threshold. The object poses are computed with scene flow within the object mask and Least-Squares estimation followed by Iterative Closest Points (ICP) for refinement. The optimization process is divided into 3 stages as follows. First, the background NeRF is initialized by sampling outside the robot arm mask and the object mask. Second, the foreground object VolSDF is initialized by sampling only within the object mask and the object poses are fixed. Finally, the neural radiance fields and the object poses are jointly optimized.

C. Memorizing the 3D objects

The next step following the reconstruction is to memorize the 3D objects so that they can be rapidly recognized on new test images. Here, we use the PVNet [4] to demonstrate how to learn an object pose estimator based on the reconstructed object model. Taking an RGB image as input, PVNet predicts the 2D keypoint positions using pixel-wise voting and computes the object pose with a Perspective-n-Point (PnP) solver [42]. As shown in Fig. 2, the training images for the PVNet are obtained by rendering the reconstructed model at a large number of object poses. At inference time, ICP is used to refine the predicted object pose by aligning the reconstructed object model and the point cloud back-projected from the depth image to improve the object pose accuracy.

D. Applications

The perception of objects can be applied to many downstream tasks. Here, we use robotic grasping as an example. To grasp an object with a gripper, the relative pose between the gripper and the base of the arm is computed as follows:

$$T_{gb} = T_{go} T_{oc} T_{cb}, \tag{12}$$

where T_{go}, T_{oc}, and T_{cb} are the relative poses between the gripper and the object, the object and the camera, and the camera and the base of the arm, respectively. As shown in Fig. 3, given the reconstructed object model, we use the analytic method Graspit! [16] to compute T_{go} and PVNet [4] to estimate T_{oc}. T_{cb} is obtained via hand-eye calibration. The details can be found in the supplementary material.

E. Implementation Details

Poking. We choose to perform 4 poking actions for each object as we empirically find this number enough to observe objects in sufficient views to obtain a complete perception. Other details of the poking process are in the supplementary material.

Reconstruction. During reconstruction, we use a batch size of 1024 rays, each sampled at 192 coordinates uniformly. 2 Adam optimizers with the learning rates decaying from 1e-3 and 5e-4 are used for the object poses and the neural radiance field parameters, respectively. The 3 stages cost 10000, 10000, and 50000 iterations, respectively. The loss weights are set to $w_1 = 1$,$w_2 = 1$,$w_3 = 0.1$, $w_4 =$ 2e-5, and $\mathbf{w}$ is set to 200.

Memorization. We synthesize 10000 images to train the PVNet. The object poses are sampled over 30 semi-spheres with different distances to the object. The background images are selected from the ScanNet dataset [43]. To increase the generalization ability of the PVNet, both the synthesized images and the images in the recorded video are used during training.

Grasping. The grasp poses are generated by the Graspit! [16] simulator and the one orienting downward is selected for real-world grasping to avoid collision between the gripper and the plane.

IV. EXPERIMENTS

A. Data collection

We capture a real-world RGB-D video to evaluate our method, where a cat, a duck, and a coffee box are put on a table. The video consists of 665 frames. To increase efficiency, we drop the frames with no moving objects, resulting in a 166-frame video. The image resolution is 1344×648. The ground-truth models for the cat and the duck are provided by the LINEMOD dataset [44], while the coffee box is represented by a cube with manually-measured sizes. The ground-truth object poses are obtained by aligning the object models with the RGB-D point clouds. A mesh

renderer is used to produce the ground-truth segmentation masks with the ground-truth object poses and the object models. We recommend watching the supplementary video for the collected data.

B. Object reconstruction evaluation

We use MaskFusion [45] as the baseline for object reconstruction. Since [45] cannot perform instance segmentation for unseen objects, we use the initialized masks introduced in Sec. III-B as the masks for them.

Tab. I compares our method with the baseline in terms of object pose accuracy. We report the mean and maximum of rotation and translation errors. Our method outperforms the baseline by a large margin, particularly in the maximum rotation errors for the cat and the duck, where we improved by about 20 degrees. Our method jointly optimizes the object poses and segmentation masks for all frames, eliminating accumulated error even for textureless objects, which is not possible with ICP used in [45].

Object	Method	Rotation (degree)	Translation (cm)
cat	MF	11.914 / 30.074	1.676 / 4.684
	Ours	**4.391 / 8.003**	**0.452 / 1.168**
box	MF	2.060 / **3.948**	0.712 / **1.452**
	Ours	**1.569** / 4.282	**0.596** / 1.716
duck	MF	14.144 / 31.871	3.728 / 8.212
	Ours	**4.070 / 12.743**	**1.116 / 3.388**

TABLE I

OBJECT POSE COMPARISON BETWEEN MASKFUSION (MF) AND OURS. WE REPORT MEAN ERROR / MAXIMUM ERROR OVER THE ENTIRE VIDEO.

Object	Method	C.D. ↓	F-score ↑	N.C. ↑	Mask IoU ↑
cat	MF	0.173	0.836	0.579	0.708
	Ours	**0.051**	**0.926**	**0.818**	**0.839**
box	MF	0.705	0.783	0.657	0.762
	Ours	**0.051**	**0.937**	**0.823**	**0.790**
duck	MF	0.177	0.812	0.587	0.674
	Ours	**0.035**	**0.963**	**0.854**	**0.771**

TABLE II

3D GEOMETRY COMPARISON BETWEEN MASKFUSION (MF) AND OURS. C.D. IS CHAMFER DISTANCE. N.C. REPRESENTS NORMAL CONSISTENCY.

Tab. II and Fig. 4 compare the results of object reconstruction and segmentation masks between our method and the baseline. Our method outperforms the baseline in all metrics and produces higher-quality segmentation masks, especially for the cat and the duck. This improvement is due to the joint optimization of object geometry and object pose, leading to globally consistent results.

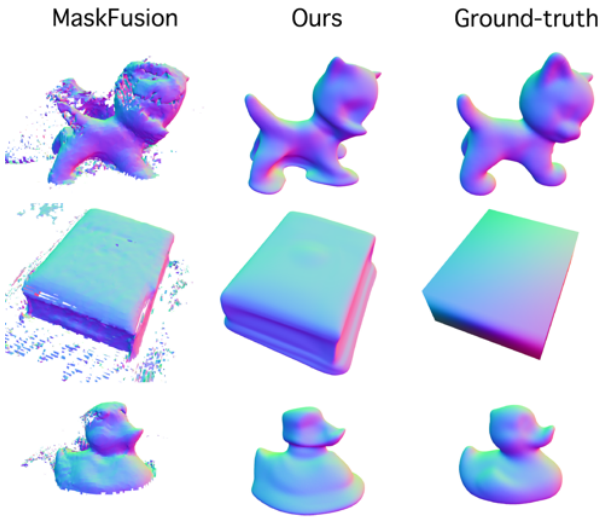

Fig. 4. **Qualitative comparison between MaskFusion and the proposed method.** The color indicates surface normal.

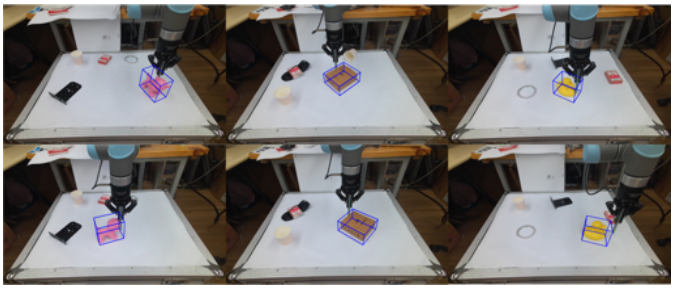

Fig. 5. **Qualitative results of object pose estimation on new test images.** The estimated bounding boxes are shown in blue. Please refer to the supplementary video for more visualization results.

C. Object memorization evaluation

To evaluate object memorization, we perform object pose estimation using the trained PVNet on new test images. Some visualization results are shown in Fig. 5, where the PVNet precisely estimates the object poses.

D. Real-world grasping

We perform a real-world robotic grasping task using a parallel gripper to grasp objects placed on a plane. The results, depicted in Fig. 6, show that the cat and the duck are successfully grasped. Due to its size, the coffee box could not be grasped from the top and is not included in the demonstration.

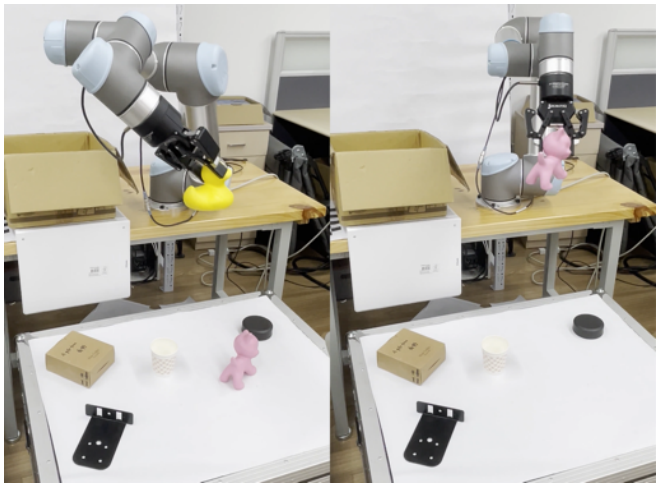

Fig. 6. **Real-world grasping of the cat and the duck.** Please refer to the supplementary video for the entire grasping process.

	Rotation (degree)	Translation (cm)
full	4.390 / 8.003	0.452 / 1.168
w/o stage-wise training	18.421 / 44.382	3.820 / 9.000
w/o foreground sampling	9.417 / 30.385	1.056 / 4.160

TABLE III

ABLATION STUDY. WE REPORT MEAN ERROR / MAXIMUM ERROR FOR THE CAT OVER THE ENTIRE VIDEO.

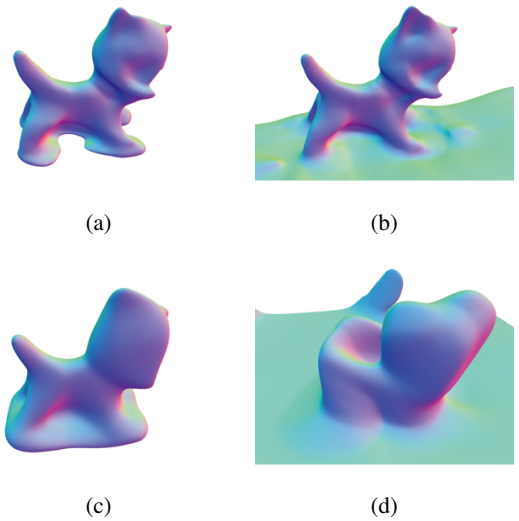

Fig. 7. **Ablation study.** Reconstructed models of the full version of the proposed method (a), without applying sparsity loss (b), without foreground sampling (c), and without stage-wise training strategy (d) are visualized.

E. Ablation

In this section, we conduct ablation experiments to analyze the effectiveness of several designs in our method. The results of the object pose evaluation and the visualization results for the cat are shown in Tab. III and Fig. 7, respectively.

The sparsity loss. To validate the benefit of the sparsity loss, we perform optimization without sparsity loss and extract the object mesh. As visualized in Fig. 7 (b), our method cannot decompose the object and the background correctly without the sparsity loss due to the motion ambiguity of the texture-poor background.

The foreground sampling strategy. To measure the effectiveness of the mask sampling strategy, we evaluate the performance of the proposed method with a random sampling strategy. As shown in the second line in Tab. III and Fig. 7 (c), the optimization could not focus on the object region and thus produce very coarse results.

The stage-wise training strategy. To measure the effectiveness of the stage-wise training strategy, we evaluate the performance of the proposed method with stage 3 only. Comparing the first line and the third line in Tab. III shows that the proposed method cannot decompose the foreground objects and the background correctly without initializing the radiance fields in stage 1 and stage 2.

V. LIMITATION

There are several directions to improve our system. First, the accuracy of depth scanning is a challenge, particularly for glossy or transparent surfaces. This leads to errors in object pose initialization and affects the accuracy of depth supervision during optimization. Second, the current reconstruction and memorization processes are time-consuming, which can be potentially addressed with faster reconstruction methods [46]–[48] and pose estimation methods that do not require training [49]–[51].

VI. CONCLUSIONS

In this paper, we proposed a new system for unseen 3D object perception. The key idea is to perform poking to discover 3D objects in the scene and then reconstruct the 3D objects based on the multi-view observations from object motion. The reconstructed models can be then utilized to train neural networks for object recognition in new test images. Our method achieved successful 3D object discovery and high-quality reconstruction in real-world scenarios, as demonstrated by experimental results. The learned neural networks can be directly applied in downstream tasks like robotic grasping, manipulation, and scene understanding. We believe that our system presents a promising approach towards the practical deployment of robots in real-world environments.

Acknowledgement. The authors would like to acknowledge the support from the National Key Research and Development Program of China (No. 2020AAA0108901) and ZJU-SenseTime Joint Lab of 3D Vision.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[3] J. Sun, L. Chen, Y. Xie, S. Zhang, Q. Jiang, X. Zhou, and H. Bao, "Disp r-cnn: Stereo 3d object detection via shape prior guided instance disparity estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 548–10 557.

[4] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.

[5] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2642–2651.

[6] G. Gkioxari, J. Malik, and J. Johnson, "Mesh r-cnn," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9785–9795.

[7] M. Runz, K. Li, M. Tang, L. Ma, C. Kong, T. Schmidt, I. Reid, L. Agapito, J. Straub, S. Lovegrove *et al.*, "Frodo: From detections to 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 720–14 729.

[8] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.

[9] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," *Advances in neural information processing systems*, vol. 29, 2016.

[10] Y. Ye, D. Gandhi, A. Gupta, and S. Tulsiani, "Object-centric forward modeling for model predictive control," in *Conference on Robot Learning*. PMLR, 2020, pp. 100–109.

[11] M. Janner, S. Levine, W. T. Freeman, J. B. Tenenbaum, C. Finn, and J. Wu, "Reasoning about physical interactions with object-oriented prediction and planning," *arXiv preprint arXiv:1812.10972*, 2018.

[12] Z. Xu, J. Wu, A. Zeng, J. B. Tenenbaum, and S. Song, "Densephysnet: Learning dense physical object representations via multi-step dynamic interactions," *arXiv preprint arXiv:1906.03853*, 2019.

[13] Z. Xu, Z. He, J. Wu, and S. Song, "Learning 3d dynamic scene representations for robot manipulation," *arXiv preprint arXiv:2011.01968*, 2020.

[14] Y. Du, K. Smith, T. Ulman, J. Tenenbaum, and J. Wu, "Unsupervised discovery of 3d physical objects from video," *arXiv preprint arXiv:2007.12348*, 2020.

[15] T. Kipf, G. F. Elsayed, A. Mahendran, A. Stone, S. Sabour, G. Heigold, R. Jonschkowski, A. Dosovitskiy, and K. Greff, "Conditional object-centric learning from video," *arXiv preprint arXiv:2111.12594*, 2021.

[16] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[17] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[18] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," *arXiv preprint arXiv:1804.05172*, 2018.

[19] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng, "Fully convolutional grasp detection network with oriented anchor box," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7223–7230.

[20] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[21] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3629–3635.

[22] Y. Qin, R. Chen, H. Zhu, M. Song, J. Xu, and H. Su, "S4g: Amodal single-view single-shot se (3) grasp detection in cluttered scenes," in *Conference on robot learning*. PMLR, 2020, pp. 53–65.

[23] T. Patten, K. Park, and M. Vincze, "Dgcm-net: dense geometrical correspondence matching network for incremental experience-based robotic grasping," *Frontiers in Robotics and AI*, vol. 7, p. 120, 2020.

[24] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.

[25] M. Van der Merwe, Q. Lu, B. Sundaralingam, M. Matak, and T. Hermans, "Learning continuous 3d reconstructions for geometrically aware grasping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 11 516–11 522.

[26] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.

[27] 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.

[28] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2018, pp. 10–20.

[29] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5231–5237.

[30] 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.

[31] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4805–4815, 2021.

[32] 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," *arXiv preprint arXiv:2106.10689*, 2021.

[33] B. Yang, Y. Zhang, Y. Xu, Y. Li, H. Zhou, H. Bao, G. Zhang, and Z. Cui, "Learning object-compositional neural radiance field for editable scene rendering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 779–13 788.

[34] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide, "Neural scene graphs for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2856–2865.

[35] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5741–5751.

[36] 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.

[37] W. Yuan, Z. Lv, T. Schmidt, and S. Lovegrove, "Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 144–13 152.

[38] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[39] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," *arXiv preprint arXiv:2002.10099*, 2020.

[40] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5752–5761.

[41] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution

3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[42] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009.

[43] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[44] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*. Springer, 2012, pp. 548–562.

[45] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2018, pp. 10–20.

[46] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *arXiv preprint arXiv:2201.05989*, 2022.

[47] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.

[48] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.

[49] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou, "Onepose: One-shot object pose estimation without cad models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6825–6834.

[50] X. He, J. Sun, Y. Wang, D. Huang, H. Bao, and X. Zhou, "Onepose++: Keypoint-free one-shot object pose estimation without cad models," *arXiv preprint arXiv:2301.07673*, 2023.

[51] I. Shugurov, F. Li, B. Busam, and S. Ilic, "Osop: A multi-stage one shot object pose estimation framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6835–6844.