

RGBGrasp: Image-based Object Grasping by Capturing Multiple Views during Robot Arm Movement with Neural Radiance Fields

Chang Liu*, Kejian Shi*, Kaichen Zhou*, Haoxiao Wang, Jiyao Zhang, Hao Dong

Abstract—Robotic research encounters a significant hurdle when it comes to the intricate task of grasping objects that come in various shapes, materials, and textures. Unlike many prior investigations that heavily leaned on specialized point-cloud cameras or abundant RGB visual data to gather 3D insights for object-grasping missions, this paper introduces a pioneering approach called RGBGrasp. This method depends on a limited set of RGB views to perceive the 3D surroundings containing transparent and specular objects and achieve accurate grasping. Our method utilizes pre-trained depth prediction models to establish geometry constraints, enabling precise 3D structure estimation, even under limited view conditions. Finally, we integrate hash encoding and a proposal sampler strategy to significantly accelerate the 3D reconstruction process. These innovations significantly enhance the adaptability and effectiveness of our algorithm in real-world scenarios. Through comprehensive experimental validation, we demonstrate that RGBGrasp achieves remarkable success across a wide spectrum of object-grasping scenarios, establishing it as a promising solution for real-world robotic manipulation tasks. The demo of our method can be found on: <https://sites.google.com/view/rgbgrasp>

I. INTRODUCTION

Robotic object grasping is a pivotal challenge in robotics due to the diverse morphology and material properties of real-world objects. The task necessitates accurate 3D geometry estimation, particularly for subsequent grasping pose determination. Although point-cloud sensors are prevalent in current approaches [1], [2], they encounter difficulties with transparent and specular objects, leading to unreliable 3D environmental information capture in these scenarios. Additionally, point clouds are often associated with high costs and provide low-resolution 3D observations.

In light of the aforementioned considerations, there has been a growing interest in leveraging the RGB modality to enhance the process of object grasping. One avenue of research focuses on achieving 3D scene perception from a single image. For instance, [3] employs a single RGB image for the estimation of 3D object detection and pose, while [4] utilizes a single image to predict 2D keypoints along with their corresponding depth information.

Besides the straightforward rationale underpinning single-image-based grasping models, the task of inferring 3D geometric information from a single image poses inherent

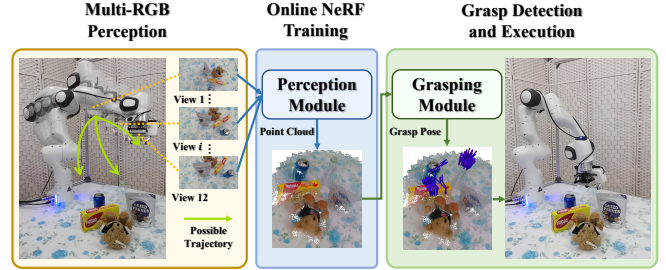


Fig. 1. **Overview of RGBGrasp.** We introduce a novel approach capable of reconstructing the 3D geometric information of a target scene using views acquired during standard grasping procedures. Our method is not limited to fixed viewpoints and can flexibly work with partial observations in different trajectories based on the environmental requirements.

challenges, notably due to the absence of scale information [5]. Consequently, contemporary approaches predominantly harness multi-view RGB data to reconstruct comprehensive 3D scenes. For instance, [6] leverages the neural radiance field (NeRF) [7] framework to aggregate information from multiple viewpoints, thereby achieving precise object grasping for items exhibiting diverse material properties.

Despite the impressive achievements of contemporary multi-view-based grasping algorithms, they encounter persistent challenges. Firstly, accurately reconstructing explicit geometric representations for 3D scenes that incorporate objects with varying material properties, such as transparent and specular objects, remains a difficult task. Secondly, these algorithms typically necessitate an unnatural multi-view data collection process, exemplified by [6] requiring a series of frames captured by orbiting around the target scene. Thirdly, multi-view-based methods typically entail extended learning durations, limiting their applicability in real-world scenarios.

Given the challenges faced by prior algorithms, we introduce RGBGrasp, an innovative multi-view grasping algorithm aimed at reconstructing high-resolution 3D target scenarios. In our configuration, an eye-on-hand camera captures multiple views of the scene as the gripper approaches the object during manipulation. Although these views exhibit minor disparities, our objective is to attain both precise scene reconstruction and efficient reconstruction speed. Precisely, we integrate prior geometric information from pre-trained monocular depth estimation networks with the 3D reconstruction capabilities of neural radiance fields to achieve precise 3D scene reconstruction in scenarios with limited views. Additionally, we utilize hash encoding and a novel sampling proposal network to enhance the expediency of the learning process

Chang Liu, Jiyao Zhang and Hao Dong are with CFCs, School of CS, Peking University and National Key Laboratory for Multimedia Information Processing. Kejian Shi is with Imperial College London. Kaichen Zhou is with Oxford University. Haoxiao Wang is with Tianjin University of Technology. Jiyao Zhang is also with Beijing Academy of Artificial Intelligence (BAAI).

* indicates equal contribution

Corresponding to hao.dong@pku.edu.cn

for targeted 3D scenarios. In summary, our contributions can be encapsulated as follows:

- We introduce depth rank loss [8], aided by a trained depth estimation model, enabling precise geometric estimation under sparse view scenarios.
- We integrate a hash-encoding strategy with a proposal sampler strategy, resulting in accelerated 3D scenario reconstruction.
- We conducted experiments for both multi-view and sparse-view scenarios, encompassing grasping and perception tasks. RGBGrasp consistently exhibited superior performance when compared to recent approaches.

II. RELATED WORK

A. Robot Grasping

Robot grasping plays a crucial role in enabling robots to interact with objects in the physical world. Extensive research has been dedicated to enhancing the grasping capabilities of robots over the years. GraspNet [9] and AnyGrasp [10] are two notable examples of grasp detection networks trained to propose viable grasp poses based on the local depth geometry observed from RGB-D cameras. Additionally, GSNet [11] introduced a novel geometrical metric called graspsness, which improves the speed and accuracy of grasp detection by a large margin.

B. Neural Radiance Field

Neural radiance field (NeRF) [7] is an advanced neural network-based approach for scene representation and synthesis. It generates photorealistic views by modeling scenes as continuous 3D functions from a set of input images and corresponding camera parameters. NeRF's ability to handle complex geometry and lighting conditions makes it valuable for applications in virtual reality [12], [13], robotics [14], [15], [16], image synthesis [17], [18], and more.

C. NeRF in Robotics

Recent studies have demonstrated the efficacy of NeRFs as a promising scene representation in several domains within robotics research. Specifically, NeRF has shown potential in robot navigation [19], [20], manipulation [21], [22], object detection [23], and 6D pose tracking [24]. DexNeRF [25] proposes to construct the NeRF representation from 49 RGB images taken from different viewpoints around the scene and use volumetric rendering to produce a depth map from a top-down view for grasping. GraspNeRF [6] leverages generalizable NeRF to enable the model to directly work on novel scenes without retraining and downgrading the number of input views needed for NeRF to 6 sparse views. However, these works still require taking RGB images 360 degrees around the scene to construct the global context. These methods will fail in some scenarios when the visual horizon is limited or viewing from the back of the scene is impractical (*e.g.*, grabbing objects from a shelf). We propose RGBGrasp and build on advantages in NeRF to solve this problem when only partial observations are available.

III. METHOD

A. Problem Statement and Method Overview

As shown in Fig. 2, RGBGrasp employs an eye-on-hand configuration by strategically mounting one RGB camera on the wrist of the robot gripper. The camera continuously captures RGB images from various angles as the robot arm directly approaches the target objects during the grasping process, those RGB images are used to progressively refine depth information. Similar to Evo-NeRF [26], we assume that the objects and the robot are placed on the same planar workspace, and the objects keep rest with stable poses within the reach of the robot arm. Additionally, camera extrinsic at each step can be calculated by gripper-to-camera transform.

The robot continuously takes observation of the scene and learns about the geometry while its arm gets closer proximity to the objects. Then it predicts some grasp poses and directly executes the grasp action based on the grasp pose of the highest confidence.

Specifically, at each time step t , given the current and previous recorded RGB images $\mathcal{I} = \{I_0, I_1, \dots, I_t\}$ with their corresponding camera extrinsics $\mathcal{P} = \{P_0, P_1, \dots, P_t\}$, we incrementally train a NeRF-Based model from scratch to estimate the depth of the scene. Subsequently, we employ state-of-the-art grasp pose detection methods, like GraspNet [27], to leverage the reconstructed depth data for predicting optimal grasp poses.

B. Scene Depth Reconstruction

Since the robot arm performs a top-down grasping task and directly takes a trajectory from the higher initial position to the scene, the gripper does not need to pre-rotate around the scene for a full or half circle in order to get observation of the whole scene. Therefore, the pictures that the on-hand RGB camera captures have a much smaller angular distribution range in the horizontal plane, which means the model may not have sufficient information about the geometry of the scene. Furthermore, the continual expansion of the training dataset can introduce instability into the training process and adversely affect the accuracy of the final reconstructed scenarios. Additionally, prevailing NeRF-based grasping algorithms are often characterized by extended training times, impeding their practical applicability.

1) *NeRF Preliminaries:* NeRF [7] learns a 5D neural radiance field which maps the spatial coordinate (x, y, z) and view direction (θ, ϕ) to the volume density σ and color c . The expected color $C(\mathbf{r})$ of camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with near and far bounds t_n and t_f is:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \sigma(\mathbf{r}(t)) c(\mathbf{r}(t), \mathbf{d}) dt$$

$$\text{where } T(t) = \exp \left(- \int_{t_n}^t \sigma(\mathbf{r}(s)) ds \right) \quad (1)$$

To discretize the computation process, NeRF approximates the expected color $\hat{C}(\mathbf{r})$ as:

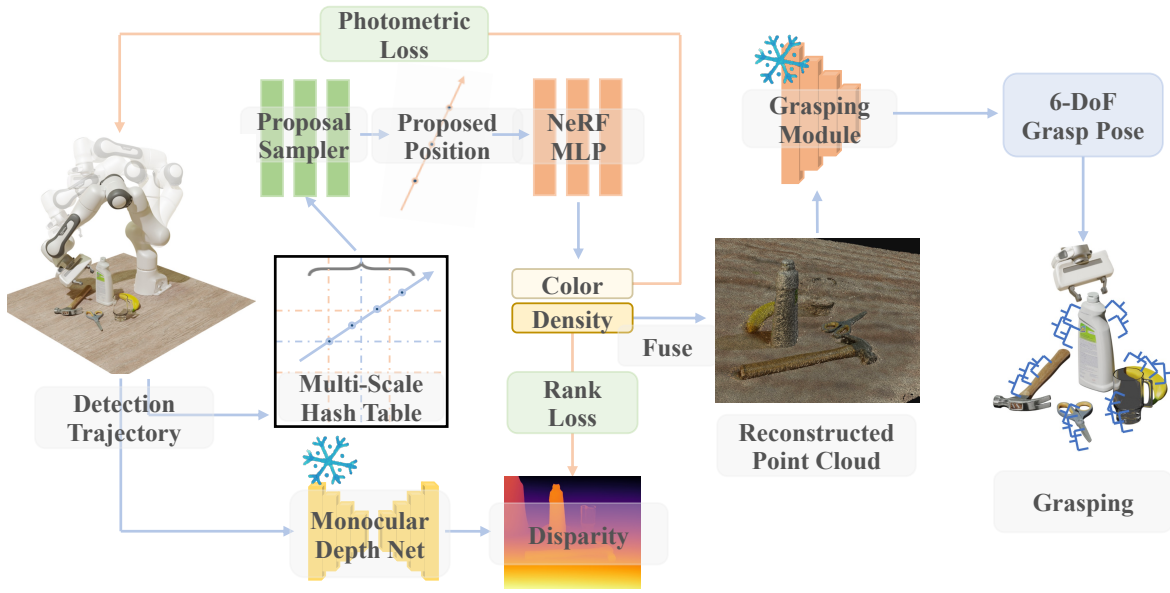


Fig. 2. **RGBGrasp Pipeline.** The robot employs an approaching trajectory to the objects when capturing multiple views to build a multi-scale hash table. Subsequently, a proposal sampler is trained to enhance the precision of sampling positions for a subsequent fine predictor. This predictor provides color and density data for individual points, where the density information contributes to the construction of the final point cloud. This resultant point cloud serves as input for a pre-trained grasping module to predict a 6-DoF grasp pose. Throughout the optimization procedure, we maintain a fixed state for the monocular depth network and the grasping module, rendering them non-trainable components. In contrast, the Hash Table, Proposal Sampler, and NeRF MLP are actively updated and subject to the learning process.

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) c_i \quad (2)$$

where $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ and $\delta_i = t_{i+1} - t_i$. Here σ_i represents the transparency of the point, and NeRF leverages alpha-compositing with alpha values $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$.

Similar to the color information computation, the depth information could be computed as:

$$Z(\mathbf{r}) = \sum_{i=1}^N \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \delta_i \quad (3)$$

2) *Geometry Regularization:* To address inaccurate depth estimation resulting from sparse views and diverse material properties of objects, RGBGrasp integrates monocular depth information \hat{Z} into the Neural Radiance Field (NeRF) learning process. However, utilizing the depth information obtained from a pre-trained monocular model directly in the training process is challenging due to the absence of scale information. To mitigate this issue, we employ the rank loss function to leverage the predicted depth information.

Despite the absence of scale in \hat{Z} , it offers precise relative depth relationships between different points. Drawing inspiration from SparseNeRF [8], RGBGrasp incorporates a rank loss to address this aspect, expressed as follows:

$$\mathcal{L}_R = \sum_{\hat{Z}_{k2} \leq \hat{Z}_{k1}} \max(Z_{k2} - Z_{k1} + \epsilon, 0), \quad (4)$$

where $k1$ and $k2$ are indices of 2D pixel coordinates, and ϵ is a small margin that allows limited depth ranking errors.

3) *Speedup NeRF training:* In pursuit of expediting the processes of training and execution, we employ the hash encoding methodology as introduced in the work by Müller et al. [28]. Within the context of a fully connected network denoted as $m(y, \Phi)$, hash encoding pertains specifically to the encoding component, denoted as $y = \text{enc}(x, \theta)$. Rather than exclusively training the network's parameters represented by Φ , we extend the training process to include the encoding weights, denoted as θ . These encoding weights encompass multiple levels of multi-resolution feature maps, totaling L levels. Each level operates independently and conceptually retains feature vectors situated at grid vertex locations. The final output is derived through the concatenation of feature maps that exhibit varying resolutions.

Furthermore, in contrast to the original NeRF methodology [7] which employs a dual-network approach consisting of a “coarse” network and a “fine” network, our approach follows the proposal sampling technique introduced in MIP-NeRF 360 [29]. Specifically, we utilize a proposal Multi-Layer Perceptron (MLP) to predict volumetric density, subsequently transformed into a proposal weight vector. These proposal weights are employed to sample intervals, which are subsequently supplied to the NeRF MLP. The NeRF MLP, in turn, predicts its own weight vector and color estimates, which are utilized in the image rendering process.

C. Grasp Pose Detection

We evaluate the geometry estimation of RGBGrasp by combining it with the downstream grasping task and analyze the grasping performance. Given the reconstructed point cloud

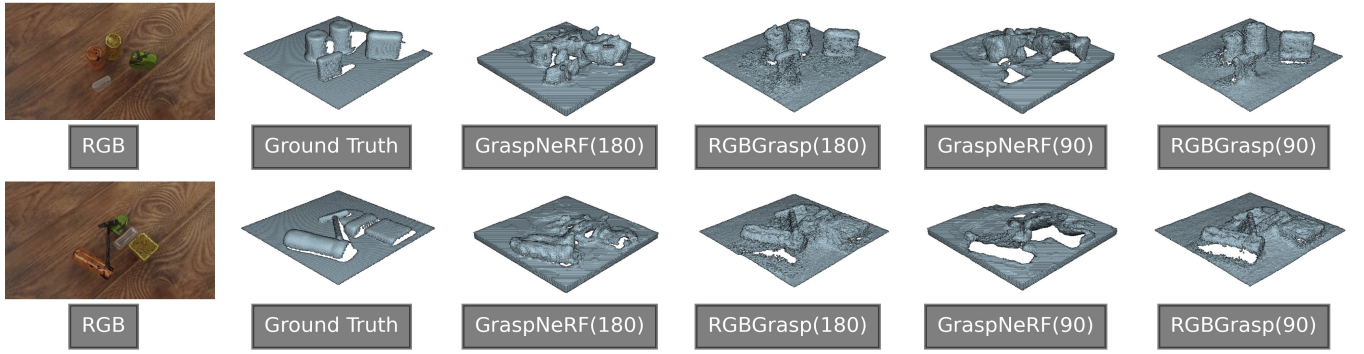


Fig. 3. **Comparison Between RGBGrasp and GraspNeRF in Terms of Reconstructed Point Clouds.** This figure presents a comparative analysis of point cloud reconstructions obtained using RGBGrasp and GraspNeRF under different trajectories, specifically, 90° and 180°. The upper row provides a comparison in the "Packed" scenario, while the lower row presents the comparison in the "Pile" scenario.

from RGBGrasp, we can pass it to the end-to-end grasping network AnyGrasp [10] to directly predict the 6-DOF grasping poses. Then the robot arm will execute the target rotation and position of the grasping pose selected from filtered pose candidates.

D. Implementation Details

1) *Robot Arm Movement Planning:* Since the camera on the robot gripper continually captures images while the robot arm approaches the scene, we find that the arm's trajectory cannot be linear from the initial position to the objects. A linear trajectory causes an extremely narrow field of view of the whole scene, which will downgrade the quality of the reconstructed point cloud to a great extent. We observe up to 4cm to 5cm estimated error for the depth of the whole scene, which is unacceptable for our grasping task.

In view of that, we design an approach strategy. The projected trajectory to the table plane (xy plane in the world coordinate system) is about a 90-degree portion of the circle, of which the center is around the scene's center. To ensure the robot arm not to get contact with the objects accidentally in advance, we set a height threshold for approaching the process of about 40cm. For each point in the trajectory, (x, y) is linearly distributed on the 90-degree portion of the circle and z is linearly distributed from the initial height and the height threshold. We only captured 12 views in total along the trajectory, with each view looking at the scene.

It should be noted that our method is **not** limited to a specific trajectory. In certain challenging scenarios, such as environments with obstacles, our method can still operate effectively within a smaller field of view than previous works. Hence, we have the flexibility to adapt the motion trajectory to different daily scenes.

2) *NeRF Training Details:* We utilize the popular NerfStudio [30] library for real-time, online training of NeRF. To enable NerfStudio's online training, we also learned from NerfBridge [31], which supports streaming the captured images on ROS to NeRF training. During training, in each step we sample 8192 rays across the batch, and the NeRF model is trained for 1200 iterations considering both the



Fig. 4. **Visualization of Trajectories with Different View Ranges.** The visualization exclusively displays the covered areas achieved through multiple views. In the original configuration (on the left (a)), views are evenly distributed to cover a full 360° at a consistent height. We successively reduce the view range to 270°, 180° and 90°, resulting in the trajectories shown in (b), (c) and (d).

reconstruction quality and the time taken to estimate the depth, with a learning rate of $6e-4$ and a weight decay of $1e-2$. Our methods are evaluated on an NVIDIA RTX 3090 GPU.

IV. EXPERIMENTS

In this section, we demonstrate the performance of our method through experiments conducted in simulation and real robot environments, respectively.

A. Experiment Setup

1) *Simulation Setup:* We train and evaluate our method in Pybullet [32], which provides a versatile and accessible platform for simulating the grasping task. All the synthetic data we used to train our method in simulation is generated by Blender [33]. We adapted the simulation setups in VGN and GraspNeRF, which constrain all the object positions within a tabletop workspace of size $30 \times 30 \times 30 \text{ cm}^3$. We simulate two types of scenes in this work. In the **pile** scene generation steps, we load the CAD models into the simulator and simulate their freefall from a point just above the desktop. We wait until none of the objects move in the scene and remove objects that fall out of the workspace. In the **packed** scene, we generate each object in an upright position on the table. The CAD models and poses of the objects left on the table are then recorded. We randomize the material of the objects to diffuse,

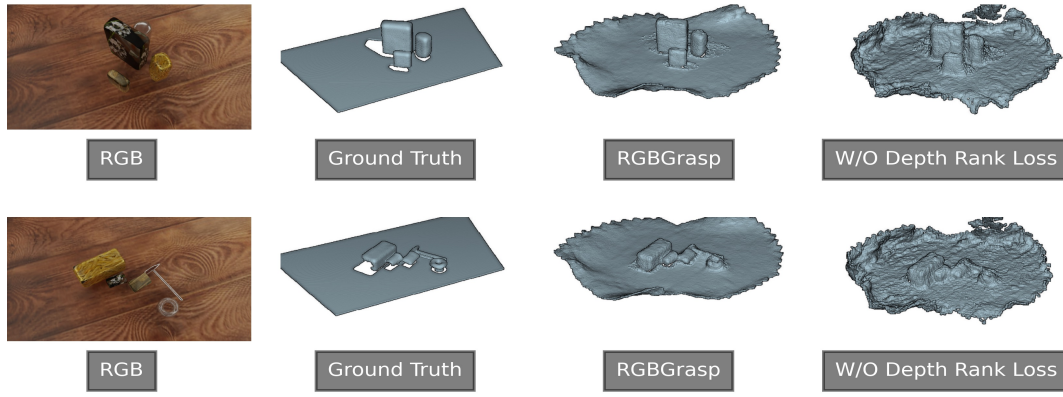


Fig. 5. **Visualization of Ablation Studies.** This figure illustrates a comparative analysis between RGBGrasp and the ablation version of our method regarding the quality of 3D geometry reconstruction.

transparent, or specular, then pass them to Blender along with their poses to generate photorealistic synthetic data.

2) *Real Robot Setup:* We attach a RealSense D415 RGB-D camera to the gripper’s wrist of the Franka Emika Panda robot arm and leverage only the RGB observation from the camera while moving to train our method. The robot arm follows our designed trajectory and dynamically predicts grasp poses in real time. Similarly, the objects are also placed in a tabletop workspace of $30 \times 30 \times 30 \text{ cm}^3$.

3) *Object Set:* To keep a fair comparison with the baseline methods, we use 56 hand-scaled object meshes from [34] dataset which is never seen by the baseline method for evaluation. We simulate these objects and randomize their textures and materials as aforementioned.

B. Baseline Methods

We compared our approach with various baseline methods that detects 6DoF grasp pose, including RGB-based methods and RGB-D-based methods.

RGB-based methods:

- **GraspNeRF [6].** An end-to-end grasping network that can deal with transparent and specular objects which utilizes a TSDF volume integrated from multi-view RGB-only images from fixed view points distributed on the circumference.
- **InstantNGP [28] + AnyGrasp [10].** We replace our NeRF model with InstantNGP, and export the estimated point cloud to AnyGrasp module to detect the grasp poses.

RGB-D-based methods:

- **Simulated raw point cloud + AnyGrasp [10].** Following the activate stereo techniques of RealSense D415 RGB-D camera, we replicate the process by rendering left and right IR images of the scene using Blender. Subsequently, employing stereo matching techniques [35], we obtain the depth information of the scene and reconstruct the single-view point cloud.

C. Evaluation Metrics

We measure our performance with the following evaluation metrics:

- **Success Rate (SR)** The number of successful grasp divided by the number of total grasp attempts
- **Declutter Rate (DR)** The percentage of removed objects among all objects in the scene. The reported declutter rates are averages over all rounds.
- **Depth RMSE** We also measure the error of depth prediction for all methods, which could be formulated as: $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$.

D. Simulation Grasping Experiments

We conduct two sets of experiments in simulation environment to illustrate the stability and effectiveness of our method. The first set of experiments shown in Sec. IV-D.1 reveals the stability of our method when the view range decreases, while the second set of experiments in Sec. IV-D.2 demonstrates the effectiveness compared to other baselines.

In all scenarios, we carry out 200 rounds of decluttering experiments, encompassing both pile (objects with random poses) and packed scenes (objects in upright position), and involving a diverse range of materials such as diffuse, transparent, and specular objects.

During each trial in a single round of experimentation, the robot arm performs a grasping action followed by the removal of one object. This process is repeated until one of the following conditions is met: (1) the entire workspace is cleared of objects, (2) the grasp detection system fails to detect a successful grasp, or (3) two consecutive grasping failures occur.

1) *Clutter Removal Experiments under Trajectories with Various View Ranges:* In this study, we evaluate the performance of RGBGrasp and GraspNeRF in the context of grasping tasks using various camera trajectories within our experimental framework. The trajectories are visualized in Fig. 4. Our findings, reported in Tab. I, indicate that RGBGrasp generally outperforms GraspNeRF in all trajectories. As the trajectory angle decreases from 360° to 90° ,

RGBGrasp maintains a high success rate without performance degradation, however degradation can be witnessed for GraspNeRF. Additionally, even in scenarios with only a 90° field of view, RGBGrasp consistently delivers competitive performance.

In this experiment, we also assess the perceptual capabilities of our proposed 3D reconstruction system in the context of grasping scenarios. Given that grasping tasks heavily rely on accurate depth perception, we conduct a comparative evaluation of our algorithm’s depth reconstruction performance against that of GraspNeRF.

The results of these comparisons are summarized in Tab. II. It is noteworthy that, in the majority of cases, our algorithm exhibits superior performance compared to GraspNeRF. An interesting observation is that GraspNeRF employs a TSDF (Truncated Signed Distance Function) as an intermediate representation for the 3D environment, resulting in a substantially higher number of generated point clouds compared to our approach. This difference in representation leads to a discrepancy between the visualizations and the quantitative results presented. Even though our algorithm demonstrates markedly improved qualitative performance in Fig. 3, the quantitative results do not reflect a substantial difference.

Qualitative results presented in Fig. 3 align with the quantitative findings reported in Tab. I.

TABLE I

RESULTS OF CLUTTER REMOVAL EXPERIMENTS IN SIMULATION WITH VARIOUS VIEW RANGES

	Pile		Packed	
	SR (%)	DR (%)	SR (%)	DR (%)
GraspNeRF (90°)	17.4	9.4	28.1	23.6
GraspNeRF (180°)	37.6	25.1	69.1	69.6
GraspNeRF (270°)	60.3	38.9	80.6	77.7
GraspNeRF (360°)	70.0	45.1	81.5	80.6
RGBGrasp (90°)	86.7	81.5	84.8	86.0
RGBGrasp (180°)	84.5	77.0	86.2	88.0
RGBGrasp (270°)	84.6	77.5	85.8	86.5
RGBGrasp (360°)	85.9	80.0	85.3	86.5

2) *Clutter Removal Experiments with Approaching Trajectory*: In this experimental setup, we conducted simulations following a predefined trajectory. Our evaluation included a comparison of RGBGrasp against several baseline methods, namely Instant-NGP with AnyGrasp as the grasp module (Instant-NGP+AnyGrasp), simulated raw point cloud with

TABLE II

DEPTH ERROR OF RECONSTRUCTION RESULTS OF CLUTTER REMOVAL EXPERIMENTS IN SIMULATION WITH VARIOUS VIEW RANGES

	Pile	Size	Packed	Size
GraspNeRF (90°)	9.61×10^{-3}	261935	1.00×10^{-2}	373179
GraspNeRF (180°)	9.05×10^{-3}	276607	9.93×10^{-3}	379318
GraspNeRF (270°)	8.40×10^{-3}	278363	8.67×10^{-3}	346129
GraspNeRF (360°)	8.07×10^{-3}	272098	8.42×10^{-3}	343121
RGBGrasp (90°)	9.94×10^{-3}	62806	9.09×10^{-3}	66967
RGBGrasp (180°)	6.58×10^{-3}	62893	7.20×10^{-3}	66973
RGBGrasp (270°)	6.89×10^{-3}	62396	6.91×10^{-3}	65922
RGBGrasp (360°)	7.12×10^{-3}	62903	7.58×10^{-3}	66974

AnyGrasp (Sim Point Cloud+AnyGrasp), and our method (RGBGrasp). GraspNeRF can only work from fixed viewpoints, and it is unable to reconstruct a reasonable TSDF volume when performing perception directly along the trajectory. Therefore, we do not compare our method with GraspNeRF in this regard.

The trajectory, as detailed in the implementation section, provided a relatively narrow 90-degree field of view, which presents a limitation for NeRF-based approaches in reconstructing fine-grained object geometry. Due to this constrained view and the trajectory’s randomized and flexible nature, Instant-NGP fails to effectively reconstruct the geometry of the scene. For instance, the point cloud is filled with artifacts such as floating points, which prevents grasp modules like AnyGrasp from providing accurate grasp poses.

We evaluate the performance in two settings: the first number in each cell represents the metric in diffuse scenes (containing only diffuse objects), and the second in mixed scenes (containing diffuse, specular and transparent objects), as presented in Tab. III. In scenes consisted of diffuse objects, the performance of RGBGrasp is comparable to RGB-D based method. For mixed scenes, accuracy of simulated depth downgrades greatly because of reflection and refraction of light, which leads to a significant decrease in SR and DR. In contrast, our method is practically unaffected in these scenes.

TABLE III

RESULTS OF CLUTTER REMOVAL EXPERIMENTS IN SIMULATION WITH VIEWPOINTS ON APPROACHING TRAJECTORY

EACH CELL: DIFFUSE/MIXED

	modality	Pile		Packed	
		SR (%)	DR (%)	SR (%)	DR (%)
Sim Point Cloud + AnyGrasp	RGB-D	83.9 /76.2	75.5/65.0	81.9 /66.1	81.5/55.0
RGBGrasp	RGB	83.7 / 82.2	81.0 / 80.0	81.5 / 80.9	82.5 / 82.0

E. Real Grasping Experiments

To further evaluate the grasping performance in real-world scenarios, we conduct 15 rounds of clutter removal experiments on both pile and packed scenes, following the protocol of simulation experiments. Each scene consists of a combination of five randomly selected objects, including objects with diffuse, transparent, and specular materials. The Franka robot arm moves toward the scene and executes the grasp predicted on the fly. The round ends with failure if two consequent grasps fails, or ends with success with all objects cleared.

We compare our RGBGrasp with two baselines, namely GraspNeRF with fixed viewpoints and AnyGrasp with monocular topdown RGB-D image as the input. Results are shown in Tab. IV. Given that each scene contains at least one transparent daily object, AnyGrasp’s performance downgrades much because of the raw, corrupted depth image of RealSense D415. However, RGBGrasp can reconstruct more accurate and fine-grained point cloud in contrast to RGB-D cameras, which boosts the final success rate of the grasping tasks.

F. Ablation Studies

For ablation analysis, we conduct clutter removal experiments on mixed scenes in simulation between RGBGrasp

TABLE IV

RESULTS OF CLUTTER REMOVAL EXPERIMENTS IN REAL WORLD

	modality	Pile		Packed	
		SR (%)	DR (%)	SR (%)	DR (%)
RealSense D415 + AnyGrasp	RGB-D	61.4 (43/70)	33.3 (5/15)	59.7 (43/72)	33.3 (5/15)
RGBGrasp	RGB	84.1 (74/88)	93.3 (14/15)	77.1 (64/83)	66.7 (10/15)

and RGBGrasp without depth rank loss.

The corresponding qualitative results are illustrated in Fig. 5, where we observe that even under these challenging conditions, our method managed to produce reasonable 3D representations of the scenes. As we can see, there are more artifacts in the reconstructed point cloud without depth supervision, especially on the boundary of the point cloud. However, most grasp modules rely on local geometry features to generate grasp poses, and these artifacts would cheat grasp modules to detect grasps around them. This phenomenon is more likely to occur in pile scenes, as objects in pile scenes have smaller dimensions and are more susceptible to artifacts. The quantitative results are shown in Tab. V.

TABLE V

ABLATION STUDIES ON CLUTTER REMOVAL EXPERIMENTS WITH VIEWPOINTS ON APPROACHING TRAJECTORY IN SIMULATION

	Pile		Packed	
	SR (%)	DR (%)	SR (%)	DR (%)
RGBGrasp w/o depth rank loss	79.3	75.5	79.7	81.0
RGBGrasp	82.2	80.0	80.9	82.0

V. CONCLUSION & FUTURE WORKS

Our algorithm pioneers accurate 3D geometry reconstruction for typical grasping trajectories by incorporating prior knowledge from a monocular depth model. To achieve real-time reconstruction, we combined hash encoding and proposal sampling. Our experiments demonstrate superior performance in various trajectories for perception and grasping tasks. However, challenges remain in certain scenarios, prompting us to explore integrating prior knowledge into object detection for enhanced accuracy in future research.

REFERENCES

- [1] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.
- [2] 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.
- [3] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmabhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis, "Single image 3d object detection and pose estimation for grasping," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3936–3943.
- [4] G. Zhai, D. Huang, S.-C. Wu, H. Jung, Y. Di, F. Manhardt, F. Tombari, N. Navab, and B. Busam, "Monograspnet: 6-dof grasping with a single rgb image," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1708–1714.
- [5] K. Zhou, L. Hong, C. Chen, H. Xu, C. Ye, Q. Hu, and Z. Li, "Devnet: Self-supervised monocular depth learning via density volume construction," in *European Conference on Computer Vision*. Springer, 2022, pp. 125–142.
- [6] Q. Dai, Y. Zhu, Y. Geng, C. Ruan, J. Zhang, and H. Wang, "Graspnerf: multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1757–1763.
- [7] 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 *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, 2020.
- [8] G. Wang, Z. Chen, C. C. Loy, and Z. Liu, "Sparsenerf: Distilling depth ranking for few-shot novel view synthesis," *arXiv preprint arXiv:2303.16196*, 2023.
- [9] 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.
- [10] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, 2023.
- [11] C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu, "Graspness discovery in clutters for fast and accurate grasp detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 964–15 973.
- [12] N. Deng, Z. He, J. Ye, B. Duinkharjav, P. Chakravarthula, X. Yang, and Q. Sun, "Fov-nerf: Foveated neural radiance fields for virtual reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 11, pp. 3854–3864, 2022.
- [13] N. Kondo, S. Kuroki, R. Hyakuta, Y. Matsuo, S. S. Gu, and Y. Ochiai, "Deep billboards towards lossless real2sim in virtual reality," *arXiv preprint arXiv:2208.08861*, 2022.
- [14] V. Blukis, T. Lee, J. Tremblay, B. Wen, I. S. Kweon, K.-J. Yoon, D. Fox, and S. Birchfield, "Neural fields for robotic object manipulation from a single image," *arXiv preprint arXiv:2210.12126*, 2022.
- [15] D. Yan, X. Lyu, J. Shi, and Y. Lin, "Efficient implicit neural reconstruction using lidar," *arXiv preprint arXiv:2302.14363*, 2023.
- [16] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6394–6400.
- [17] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "Graf: Generative radiance fields for 3d-aware image synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 154–20 166, 2020.
- [18] J. Gu, L. Liu, P. Wang, and C. Theobalt, "Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis," *arXiv preprint arXiv:2110.08985*, 2021.
- [19] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.
- [20] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.

- [21] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, “3d neural scene representations for visuomotor control,” in *Conference on Robot Learning*. PMLR, 2022, pp. 112–123.
- [22] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, “Nerf-supervision: Learning dense object descriptors from neural radiance fields,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6496–6503.
- [23] B. Hu, J. Huang, Y. Liu, Y.-W. Tai, and C.-K. Tang, “Nerf-rpn: A general framework for object detection in nerfs,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 528–23 538.
- [24] P. Chidananda, S. Nair, D. Lee, and A. Kaehler, “Pixtrack: Precise 6dof object pose tracking using nerf templates and feature-metric alignment,” *arXiv preprint arXiv:2209.03910*, 2022.
- [25] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, “Dex-nerf: Using a neural radiance field to grasp transparent objects,” *arXiv preprint arXiv:2110.14217*, 2021.
- [26] J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, and K. Goldberg, “Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects,” in *6th Annual Conference on Robot Learning*, 2022.
- [27] 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 (CVPR)*, June 2020.
- [28] 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.
- [29] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, “Mip-nerf 360: Unbounded anti-aliased neural radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.
- [30] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, “Nerfstudio: A modular framework for neural radiance field development,” in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH ’23, 2023.
- [31] J. Yu, J. E. Low, K. Nagami, and M. Schwager, “Nerfbridge: Bringing real-time, online neural radiance field training to robotics,” *arXiv preprint arXiv:2305.09761*, 2023.
- [32] E. Coumans and Y. Bai, “Pybullet, a python module for physics simulation for games, robotics and machine learning,” <http://pybullet.org>, 2016–2021.
- [33] “Blender,” <https://www.blender.org/>.
- [34] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, “Volumetric grasping network: Real-time 6 dof grasp detection in clutter,” in *Conference on Robot Learning*. PMLR, 2021, pp. 1602–1611.
- [35] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.