2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Residual-NeRF: Learning Residual NeRFs for Transparent Object Manipulation

Bardienus P. Duisterhof[1], Yuemin Mao[1], Si Heng Teng[1], Jeffrey Ichnowski[1]

***Abstract*— Transparent objects are ubiquitous in industry, pharmaceuticals, and households. Grasping and manipulating these objects is a significant challenge for robots. Existing methods have difficulty reconstructing complete depth maps for challenging transparent objects, leaving holes in the depth reconstruction. Recent work has shown neural radiance fields (NeRFs) work well for depth perception in scenes with transparent objects, and these depth maps can be used to grasp transparent objects with high accuracy. NeRF-based depth reconstruction can still struggle with especially challenging transparent objects and lighting conditions. In this work, we propose Residual-NeRF, a method to improve depth perception and training speed for transparent objects. Robots often operate in the same area, such as a kitchen. By first learning a *background NeRF* of the scene without transparent objects to be manipulated, we reduce the ambiguity faced by learning the changes with the new object. We propose training two additional networks: a *residual NeRF* learns to infer residual RGB values and densities, and a *Mixnet* learns how to combine background and residual NeRFs. We contribute synthetic and real experiments that suggest Residual-NeRF improves depth perception of transparent objects. The results on synthetic data suggest Residual-NeRF outperforms the baselines with a 46.1 % lower RMSE and a 29.5 % lower MAE. Real-world qualitative experiments suggest Residual-NeRF leads to more robust depth maps with less noise and fewer holes. Website: `https://residual-nerf.github.io`**

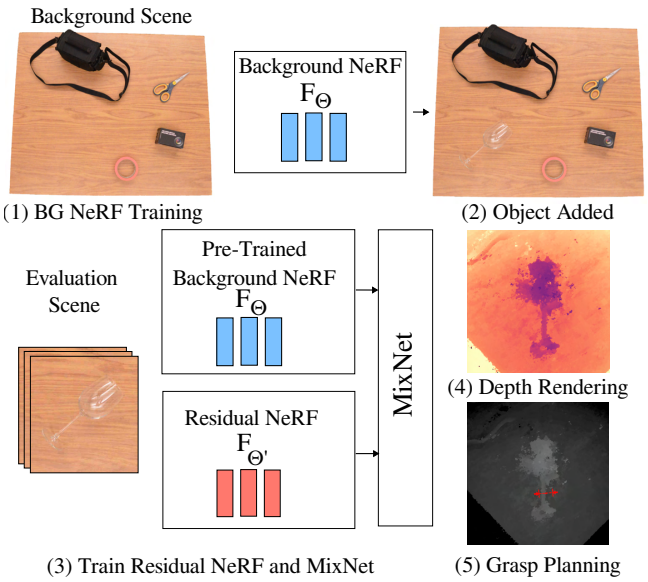


Fig. 1: Residual-NeRF, a method that leverages mostly static scenes to improve depth perception and speed up training. Residual-NeRF begins by learning a *background NeRF* of the entire scene without transparent objects. Following this, we learn a *residual NeRF* and a *Mixnet* to complement the *background NeRF*.

I. Introduction

Dexterous manipulation of transparent objects can enable robots to perform new and impactful tasks in industry, pharmaceuticals, and households. Robots often use depth images of objects to decide what action (e.g., pull, lift, or drop) to perform. However, common depth sensors struggle to capture depth images for arbitrary transparent objects [1], [2], [3], [4]. Learning-based approaches for transparent object depth estimation work well in-distribution, but can struggle to generalize outside their training data [1]. The lack of surface features on transparent objects also makes it challenging to retrieve depth maps using approaches such as COLMAP [5].

Neural radiance fields (NeRFs) [6] are implicit neural network scene representations trained on multiple views of the same scene and capable of state-of-the-art novel view synthesis. NeRF combines a multi-layer perception (MLP) with classic volumetric rendering techniques to achieve photorealistic novel view synthesis results. The only assumptions in NeRF are those imposed by traditional volumetric rendering techniques [6].

Recently, Dex-NeRF [1] and Evo-NeRF [7] showed that NeRFs are effective at depth perception of transparent objects. However, these methods also showed that NeRFs tend to struggle with particularly challenging transparent objects, such as wine glasses or kitchen foil with challenging light conditions. The challenge with transparent objects originates from a lack of features, and large view-dependent changes in appearance.

Dex-NeRF, while achieving high grasp success rates, was slow to compute. Evo-NeRF [8] significantly sped up NeRF-based grasping by *evolving* NeRFs. In this work, we extend on the Evo-NeRF motivations: (1) speed up training to a successful grasp, and (2) learning NeRFs for changing scenes, where small gradients in empty regions can be challenging [8].

We propose Residual-NeRF (Figure 1), a method to leverage a strong scene prior to improving depth perception of transparent objects. In many scenarios, the geometry of the robot's work area is mostly static and opaque, e.g., shelves, desks, and tables. Residual-NeRF leverages the static and opaque parts of the scene as a prior, to reduce ambiguity and improve depth perception. Residual-NeRF first learns a *background NeRF* of the entire scene by training on images without the transparent objects present. Residual-NeRF then uses images of the full scene with the transparent objects to learn a *residual NeRF* and a *Mixnet*. The algorithm

[1]Carnegie Mellon University, The Robotics Institute, {bduister, jeffi}@cmu.edu

adaptively mixes both NeRFs for each point along the ray, using a novel *Mixnet* multi-layer perceptron (MLP). The Mixnet infers a mixing weight between the background and residual NeRF using a location-dependent multi-resolution hash encoding.

We evaluate Residual-NeRF on nine photo-realistic synthetic and three real scenes and compare its performance to other relevant NeRF algorithms. We compare 1) learning speed, 2) depth reconstruction, and 3) robot grasp quality. The results suggest that Residual-NeRF improves on the state-of-the-art in depth mapping accuracy with a 46.1 % lower RMSE and a 29.5 % lower MAE. The results also suggest Residual-NeRF speeds up training and leads to more robust grasp planning.

In summary, we contribute:

- Residual-NeRF, a novel algorithm that learns a *residual NeRF* and *background NeRF*, mixed in 3D space by a *Mixnet* MLP.
- Experiments that suggest Residual-NeRF infers improved depth maps of transparent objects, speeds up training, and yields more robust grasp planning.
- Synthetic and real datasets.

II. Related Work

Neural Radiance Fields (NeRF): NeRF [6] is a scene representation capable of photo-realistic novel view reconstruction. It is inspired by classic volumetric rendering techniques, tracing rays through the scene to infer radiance and density using a neural network. For every point in 3D space, the NeRF MLP infers RGB radiance and density to render an RGB image. NeRF forms the basis for sensing in this paper.

After the introduction of NeRF, several works proposed improvements. A key challenge for the original NeRF algorithm [6] was the long training and inference time—often requiring several hours to train for a single scene. Recent progress has shown that novel representations and system optimizations [9], [10], [11], [12], [13], [14], [15], [9], [**?**], and depth supervision [16], [17], [18], [19], [20] can significantly speed up training. We leverage existing optimized code to speed up training [21].

Other works have focused on improving novel view synthesis performance in challenging conditions and with sparse camera views without extrinsic calibration. It is now possible to capture a NeRF of transparent objects [7], [1], reflective surfaces [22] and low-light environments [23]. Many NeRF and other works have aimed to reduce the number of views [24], [25], [26], [27], [28] and extrinsic camera calibration [29], [30], [31], [32], [28] required. Another line of work extends novel view synthesis to dynamic scenes [33], [34], [35], [36].

Merging NeRFs: Previous work has introduced methods to enable evaluating and training multiple NeRFs in a single scene. Reiser et al. [37] introduced KiloNeRF, with the contribution of using thousands of smaller MLPs ordered in a 3D grid instead of using only a single MLP. KiloNeRF maps a position $\mathbf{x}$ to a tiny NeRF using spatial binning to its 3D grid, and achieves faster inference.

Tancik et al. [38] introduced Block-NeRF, a variant of NeRF that can represent large-scale environments. Block-NeRF also contains an appearance embedding to accommodate changes in, e.g., lighting. Zhang et al. [39] introduced NeRFusion, a method that leverages TSDF-based fusion techniques for efficient large-scale reconstruction. The authors propose to infer a per-frame local radiance field via direct network inference. They are then fused using a recurrent neural network to reconstruct a global scene representation.

In summary, related work has demonstrated novel approaches to improve view reconstruction and inference speed in large scenes. Residual-NeRF aims to improve depth mapping of transparent objects for manipulation, by learning a Residual NeRF and Mixnet on top of a pretrained background NeRF. Mixnet does not require a pre-defined area for each NeRF, but learns to mix both NeRFs at every point in space.

Depth Perception of Transparent Objects: Several works have proposed methods for accurate depth perception, shape estimation, and/or pose estimation. Xie et al. [40] developed a pipeline based on transformer neural networks capable of transparent object segmentation. Phillips et al. [3] leveraged a random forest algorithm to extract the pose and shape of transparent objects. Xu et al. [4] contributed an algorithm for estimating 6-degrees-of-freedom (DOF) pose of a transparent object using only a single RGBD image. Wang et al. [41] contributed MVTrans for depth mapping, segmentation, and pose estimation of transparent objects. Chen et al. [2] contributed a benchmark dataset for segmentation, object-centric pose estimation, and depth completion.

Ichnowski et al. [1] showed how NeRFs can be leveraged to infer state-of-the-art depth perception of transparent objects, and unlike data-centric approaches, did not require a prior training on a set of objects.

III. Problem Statement

Given is a sequence of images of a scene without the transparent objects present. I_{bg_n} with $n \in [1, \ldots, N_{\text{bg}}]$ each with camera matrix P_n (i.e., the intrinsics and extrinsics). In addition, we also have access to images of the same scene with one or more transparent objects present, I_{res_n}, with $n \in [1, \ldots, N_{\text{res}}]$, P'_n. P'_n and P_n are not necessarily the same, N_{bg} and N_{res} are also not necessarily equal.

The objective is to recover novel view depth maps from any given (depth) camera pose P in the scene, and then use the novel views for grasp planning. As with Dex-NeRF and Evo-NeRF, the goal is to facilitate using a grasp planner such as Dex-Net [42]. A high-quality depth map of both opaque and transparent objects in the scene will aid motion planning and the selection of grasp location. Perceived holes in objects can lead to collisions, whereas hallucinating non-existant surfaces may lead to no available grasp location.

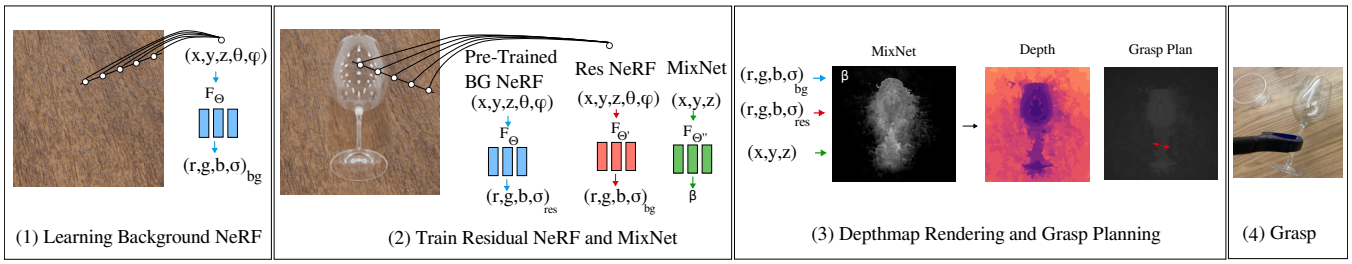


Fig. 2: Residual-NeRF, a method that leverages mostly static scenes to improve depth perception and speed up training. We first learn a *background NeRF* of the scene without transparent objects and leverage it as a scene prior. Following this, we learn a *residual NeRF* and *Mixnet*. The *Mixnet* is an MLP that learns to combine the *background NeRF* and the *residual NeRF*. Equation 5 describes how the output of the *Mixnet* MLP is used to combine the two NeRFs.

IV. Method

Residual-NeRF, shown in Figure 2, recovers depth using multiple camera views by first learning a background NeRF, then training a Mixnet and a residual NeRF (Section IV-C). As it builds on NeRF, we first review prelimaries of novel view synthesis with NeRF (Section IV-A). Once trained, we then use the NeRFs to render depth maps (Section IV-B) for the grasp planner.

A. Preliminary: Training NeRF

NeRFs [6] trace rays through images of the scene to train an MLP, Φ_r, to infer radiance and density $(RGB\sigma)$ for an input world coordinate and viewing direction (x, y, z, θ, ϕ). To accommodate high-frequency changes in 3D space, the input to NeRF is concatenated with a positional encoding in the form of sinusoids with varying frequencies. Some recent NeRF implementations use alternative representations such as a multi-resolution hash grid [9] encoding to speed up training. Thus, $\Phi_r(h(x, y, z, \theta, \phi)) = (RGB\sigma)$, where $h(\cdot)$ is the encoding function.

The expected color $C(\mathbf{r})$ by volumetric rendering is in Equation 1.

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \tag{1}$$

with

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

$T(t)$ is the accumulated transmittance along the ray from t_n to t. NeRFs approximate the function $C(\mathbf{r})$ by a learned function $\hat{C}(\mathbf{r})$, computed using Equation 2.

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i, \tag{2}$$

with

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right),$$

Here $\delta_i = t_{i+1} - t_i$, i.e., the distance between samples along the ray. The MLP learns $\mathbf{c}_i$, the color emitted at each point on the ray, and σ_i, the density at each point on the ray.

A NeRF can be trained by setting the loss to the RMSE between rendered and GT images. Equation 3 defines this photometric loss.

$$\mathcal{L}_{\text{pho}} = \sum_{(i,\mathbf{r})\in\Omega_r} ||\hat{C}_i(\mathbf{r}) - C_i(\mathbf{r})||_2^2, \tag{3}$$

Here $i \in [1, ..., N]$ is the frame number, $\mathbf{r}$ is the pixel location, and Ω_r is the set of all pixel locations across all frames.

B. Depth from NeRF

We adopt the approach for depth perception of transparent objects proposed by Ichnowski et al. [1]. Prior works propose computing depth using Equation 4,

$$\hat{D}(\mathbf{r}) = \sum_{i}^{N} T_i(1 - \exp(-\sigma_i\delta_i))t_i, \tag{4}$$

When propagating rays through transparent objects, this equation proves to be inaccurate as a result of multiple surfaces radiating light along the ray. Instead, as proposed by Ichnowski et al. [1], we find the point closest to the camera for which $\sigma_i \geq m$, where m is a tunable parameter. Prior work suggested that m was not overly sensitive and that a single tuned value could be reused across multiple scense and lighting conditions.

C. Learning a Residual NeRF

In this work, we propose a method to leverage a NeRF of the background scene (i.e., the scene without the transparent object) to improve 3D reconstruction. Equation 5 shows the modification.

$$\hat{C}(\mathbf{r}) = \textstyle\sum_{i=1}^{N} T_i(1 - \exp(-((1-\beta)\sigma_{\text{bg}} + \beta\sigma_{\text{res}})_i\delta_i))(\mathbf{c}_{\text{bg+res}})_i, \tag{5}$$

with

$$\begin{aligned}
\mathbf{c}_{\text{bg+res}} &= S((1.0 - \beta)\mathbf{c'}_{\text{bg}} + \beta\mathbf{c'}_{\text{res}}),\\
(\mathbf{c'}_{\text{res}}, \sigma_{res}) &= \Phi_{\text{res}}(h(x, y, z, \theta, \phi)),\\
(\mathbf{c'}_{\text{bg}}, \sigma_{bg}) &= \Phi_{\text{bg}}(h(x, y, z, \theta, \phi)),\\
\beta &= S(\Phi_{\text{mix}}(h(x, y, z))),\\
S(x) &= \frac{1}{1 + e^{-x}},
\end{aligned}$$

For every point along the ray, MLP Φ_{mix} takes in the encoded position and infers $\beta \in [0, 1]$. The background

(1) Synthetic Objects in Blender (2) Real-World Scenes

Fig. 3: The scenes used for evaluation. We create nine synthetic Blender [43] scenes with transparent objects and three real-world scenes, divided into scenes A-C with increasing difficulty.

NeRF infers σ_{bg} and $\mathbf{c'}_{\text{bg}}$ by training on a set of images of the background without the transparent object, while the residual NeRF infers σ_{res} and $\mathbf{c'}_{\text{res}}$ by training with the transparent object in the scene. The vector $\mathbf{c'}$ is the color $\mathbf{c}$ before it is passed through its sigmoid activation. We add the background and residual NeRF values before going through their final activation layers, such as the sigmoid. The background NeRF is not updated while training the residual NeRF.

V. Experiments

We evaluate Residual-NeRF against the baselines (Section V-B) on synthetic Blender scenes (Section V-C) with ground truth depth. We also present a qualitative evaluation against the baselines on real-world scenes (Section V-G). The results suggest Residual-NeRF outperforms the baselines by generating higher-quality depth maps. Finally we show Residual-NeRF speeds up training (Section V-H), can improve grasp planning using Dex-Net (Section V-I), and we ablate the impact of the Mixnet (Section V-J).

A. Hyperparameters

NeRFs require the setting of numerous hyperparameters potentially with great impact on performance. We have kept all vanilla NeRF parameters constant between runs, e.g., scene area, scale, and adaptive ray marching turned off. We set m (Section IV-B) to 3.0 for all Blender scenes evaluations for all methods, and tuned to the best of our ability for each real environment and method. We tune m by starting from 0.0, to then increase m until no more holes in the depth maps disappear. From this point, increasing m will lead to more noise and no additional gains.

B. Baselines

As baselines, we evaluate Residual-NeRF against NeRF [6] and Dex-NeRF [1]. While other multi-view stereo (MVS) methods for transparent objects exist, to the best of our knowledge, they do not accept arbitrary poses.

C. Synthetic Blender Data

Residual-NeRF cannot be evaluated on existing datasets such as ClearPose [2], which captures 63 transparent objects, due to the lack of background images for training the background NeRF. Therefore, we use Blender [43] to render nine photo-realistic objects as shown in Figure 3. We use the poses in a hemisphere from the original NeRF datasets [6] to render train, test, and validation images. Background NeRF and residual NeRF receive images taken from the same 100 train poses.

D. Real World Data

We take images of transparent objects on a wooden table to evaluate Residual-NeRF in the real world. We take the images using a Nikon D3200 DSLR camera, the poses are taken in a random hemisphere. Lighting conditions, focal length, aperture, ISO, and white balance were constant while collecting datasets. We use COLMAP [5] to retrieve the camera poses.

The background NeRF receives 100 images to create a strong scene prior. The residual NeRF, Mixnet, and the baselines are trained with 50 images, excluding those omitted by COLMAP. This experiment design simulates scenarios where a predominantly static scene can be extensively observed from many poses, but the scene under evaluation is limited to a smaller number of viewpoints.

The three evaluation scenes A-C in increasing difficulty are visible in Figure 3. Scene A has four opaque objects in the background scene, with an added transparent coffee container with coffee in the evaluation scene. Scene B has the same four opaque objects in the background scene but with a wine glass added in the evaluation scene. Scene C has 6 transparent objects present in the background in the background scene, 3 more transparent objects are added in the evaluation scene: a wine glass, kitchen wrap, and a glass bottle with a blue cap.

E. Implementation Details

We implement Residual-NeRF and the baselines by building on Torch-NGP [21]. Torch-NGP [21] is a PyTorch version of InstantNGP [9], and uses a multi-resolution hash grid to speed up training. It runs at a similar but slightly slower speed as compared to Instant-NGP while offering a more efficient development cycle with only raytracing implemented in CUDA. We train Residual-NeRF and all baselines on an RTX 4090 and an AMD Ryzen Threadripper PRO 5965WX with 256 GB of RAM.

F. Blender Depth Results

We evaluate the inferred depth maps by Residual-NeRF and the baselines by comparing them against the ground truth provided by Blender.

1) Quantitative Comparison: We compare Residual-NeRF against NeRF and Dex-NeRF by computing the MAE (Equation 6) and RMSE (Equation 7).

$$\text{MAE} = \frac{\sum_{(i,\mathbf{r})\in\Omega_r} \|\hat{D}_i(\mathbf{r}) - D_i(\mathbf{r})\|_1}{n}, \tag{6}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{(i,\mathbf{r})\in\Omega_r} \|\hat{D}_i(\mathbf{r}) - D_i(\mathbf{r})\|^2}{n}}, \tag{7}$$

Here $i \in [0, ..., N]$ is the frame number, $\mathbf{r}$ is the pixel location, and Ω_r is the set of all pixel locations across frames.

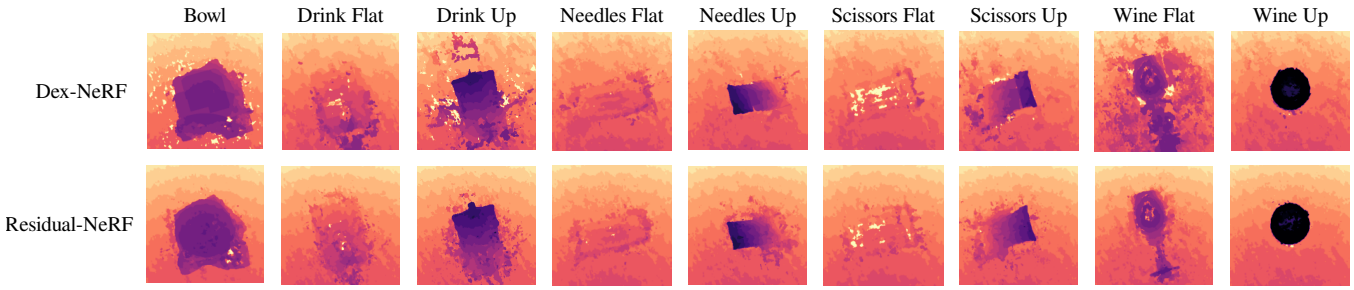


Fig. 4: Depth maps for Residual-NeRF and Dex-NeRF evaluated on the synthetic Blender dataset. The results suggest Residual-NeRF improves depth maps with fewer holes and less noise.

TABLE I: Root Mean Square Error (RMSE) in synthetic Blender Scenes.

	Bowl	Drink Flat	Drink Up	Needles Flat	Needles Up	Scissors Flat	Scissors Up	Wine Flat	Wine Up
Method					RMSE ↓				
NeRF [6]	0.1900	0.6398	0.4361	0.3529	0.1599	0.5523	0.1622	0.2166	0.5973
Dex NeRF [1]	0.0365	0.1065	0.0699	0.0293	**0.0431**	0.2624	0.0320	0.0425	0.3159
Residual-NeRF	**0.0213**	**0.0234**	**0.0316**	**0.0208**	0.0437	**0.0506**	**0.0289**	**0.0388**	**0.2462**

TABLE II: Mean Absolute Error (MAE) in synthetic Blender Scenes.

	Bowl	Drink Flat	Drink Up	Needles Flat	Needles Up	Scissors Flat	Scissors Up	Wine Flat	Wine Up
Method					MAE ↓				
NeRF [6]	0.1453	0.3424	0.3062	0.1698	0.1067	0.2505	0.1103	0.1483	0.2157
Dex NeRF [1]	0.0195	0.0189	0.0255	0.0127	**0.0193**	0.0302	0.0160	0.0255	0.0338
Residual-NeRF	**0.0140**	**0.0138**	**0.0170**	**0.0115**	0.0207	**0.0129**	**0.0140**	**0.0142**	**0.0238**

$\hat{D}(\mathbf{r})$ is the inferred depth in meters, $D(\mathbf{r})$ is the GT depth in meters. We crop each image before evaluation to focus evaluation on the transparent object. In each crop, the entire transparent object is visible, while the background is partially cropped out.

Table I shows the RMSE and Table II the MAE for Residual-NeRF compared against the baselines. It shows that Residual-NeRF outperforms the baselines except for the 'Needles Up' scene.

2) Qualitative Comparison: Figure 4 shows the depth maps inferred by Dex-NeRF [1] and Residual-NeRF. For most scenes, it shows that Residual-NeRF reduces holes and noise in depth maps. Intuitively, the background NeRF makes it less likely for holes in the table to appear, as a result of the scene geometry prior.

The bowl, drink up, and wine flat scenes show that transparent objects can make reconstruction of opaque objects more difficult, as a result of the introduced ambiguity. Holes in opaque objects may cause the end effector to collide with unseen objects.

The results could be further improved by tuning m (Section IV-B) for each scene, we have opted for setting $m = 3$ for all scenes. In our testing, the optimal m appears to be scene-dependent as well as NeRF implementation-dependent. Ichnowski et al. [1] found $m = 15$ to work best, evaluating different scenes and using a NeRF implementation without multi-resolution hash encoding.

G. Real World Depth Results

Figure 5 shows the depth maps inferred by Dex-NeRF [1] and Residual-NeRF in the real world. The results show Residual-NeRF improves the depth maps using the background NeRF as a scene prior. Similar to the results in simulation, we observe fewer holes in the table and in the objects themselves.

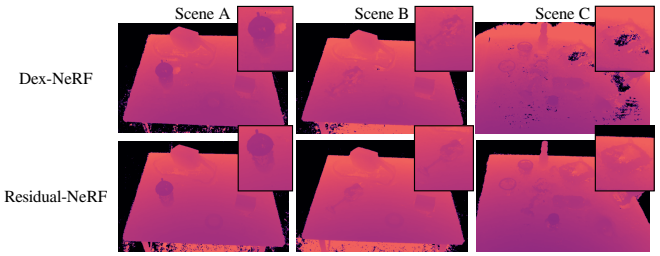


Fig. 5: Depth maps inferred by Dex-NeRF [1] and Residual-NeRF in the real world.. The result suggest Residual-NeRF results in fewer holes and less noise.

H. Training Speed

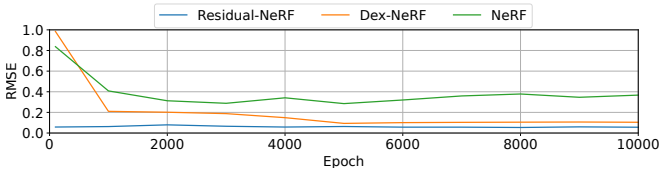
Speeding up NeRF training is critical to making NeRF a viable option in industrial and home applications. Any additional speed-ups provided by Residual-NeRF can further improve sped-up implementations such as Evo-NeRF [8].

To evaluate the quality of depth reconstruction over time, we log depth maps along with the elapsed training epochs. The resulting RMSE and MAE averaged over all synthetic scenes during training is visible in Figure 6, assuming a pre-trained background NeRF for Residual-NeRF.

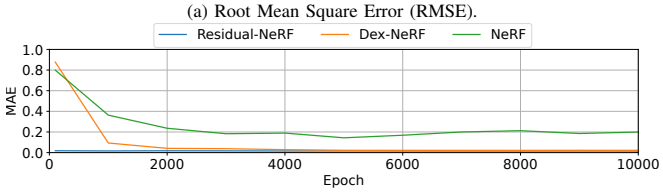
The results suggest Residual-NeRF successfully leverages the background NeRF to speed up training. We implement all methods in the same Torch-NGP [21] codebase, making implementation-related differences in training speed unlikely.

I. Grasp Planning using Dex-Net

Figure 7 shows the output of Dex-Net on Residual-NeRF and Dex-NeRF. The results suggest the output from Residual-NeRF leads to more robust grasping. We use the grasp



(a) Root Mean Square Error (RMSE).



(b) Mean Absolute Error (MAE).

Fig. 6: RMSE and MAE in depth maps, logged over the number of training epochs and averaged over all synthetic scenes. The results suggest Residual-NeRF greatly improves training speed with respect to depth reconstruction.

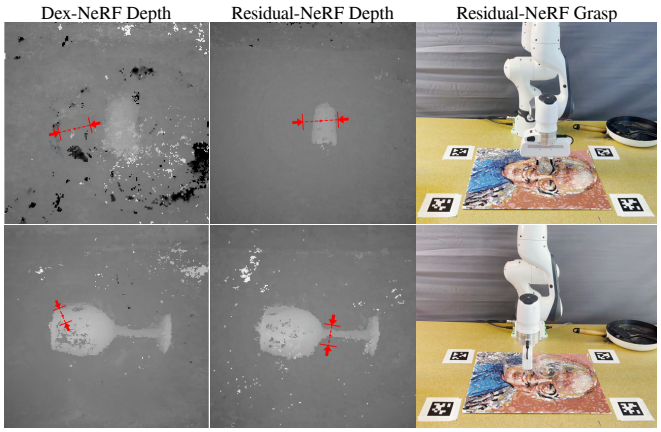


Fig. 7: Dex-Net evaluated on the output of Dex-NeRF and Residual-NeRF. The results suggest the output from Residual-NeRF leads to more robust grasping.

location inferred by Dex-Net to grasp transparent objects using a Franka robot manipulator and its parallel-jaw gripper.

J. Residual-NeRF-S Mixnet Ablation

To better understand the use of the Mixnet, we evaluate an alternative method where NeRFs are merged by adding. We refer to this method as Residual-NeRF-S. The equations for Residual-NeRF-S are:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-(\sigma_{\text{bg}} + \sigma_{\text{res}})_i \delta_i))(\mathbf{c}_{\text{bg+res}})_i, \quad (8)$$

with

$$\begin{aligned} \mathbf{c}_{\text{bg+res}} &= S(\mathbf{c'}_{\text{bg}} + \mathbf{c'}_{\text{res}}), \\ (\mathbf{c'}_{\text{bg}}, \sigma_{bg}) &= \Phi_{\text{bg}}(h(x, y, z, \theta, \phi)), \\ (\mathbf{c'}_{\text{res}}, \sigma_{res}) &= \Phi_{\text{res}}(h(x, y, z, \theta, \phi)), \\ S(x) &= \frac{1}{1 + e^{-x}}, \end{aligned}$$

Instead of training a Mixnet to merge background and residual NeRFs, the outputs of both MLPs are simply added before their activation layer. Figure 8 shows a comparison of depth and view reconstruction, both are worse compared to Residual-NeRF.

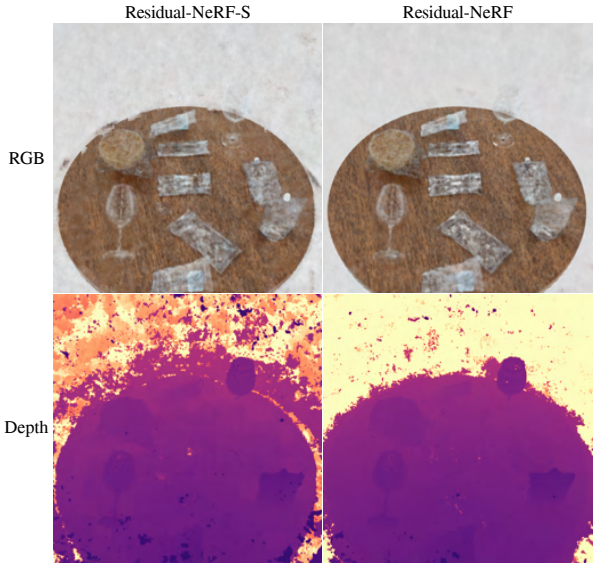


Fig. 8: Naive Merging (Residual-NeRF-S) as introduced in Equation 8 compared against Mixnet (Residual-NeRF). The results suggest Mixnet helps view reconstruction and depth mapping when merging NeRFs.

Especially with partially opaque objects, significant noise is introduced in the view reconstruction and depth maps. We believe this is due to the unpredictable behavior of the background NeRF in areas that were previously unoccupied. Without Mixnet, the residual NeRF not only has to understand the scene but also deduce the background NeRF's color and density predictions in previously empty regions.

VI. Conclusion and Discussion

In this work, we introduce Residual-NeRF, a method designed to leverage a strong prior on the background scene to improve depth perception and speed up training. We study the problem of perceiving depth in mostly static scenes with transparent objects, as is typical of a robot's working area, e.g., shelves, desks, and tables.

Residual-NeRF begins by learning a *background NeRF* of the entire scene without transparent objects. Following this, a *residual NeRF* and *Mixnet* are trained to complement the *background NeRF*. The results suggest Residual-NeRF learns faster and achieves better depth mapping, compared to the baselines.

This work could be improved by comparing against more MVS methods non-specific to transparent objects. Future work may also include combining Residual-NeRF with recent advances in depth map completion. Future research could explore the performance across different transparent objects and environment conditions.

VII. Acknowledgement

We thank the Center for Machine Learning and Health (CMLH) for their generous Fellowship in Digital Health. We also thank the Pittsburgh Supercomputing Center for their resources, and Uksang Yoo, Peter Schaldenbrand and Jean Oh for help with the robot experiments.

REFERENCES

[1] J. Ichnowski*, Y. Avigal*, J. Kerr, and K. Goldberg, "Dex-NeRF: Using a neural radiance field to grasp transparent objects," in *Conference on Robot Learning (CoRL)*, 2020.

[2] X. Chen, H. Zhang, Z. Yu, A. Opipari, and O. C. Jenkins, "Clearpose: Large-scale transparent object dataset and benchmark," in *European Conference on Computer Vision*, 2022.

[3] C. Phillips, M. Lecce, and K. Daniilidis, "Seeing glassware: from edge detection to pose estimation and shape recovery," 06 2016.

[4] C. Xu, J. Chen, M. Yao, J. Zhou, L. Zhang, and Y. Liu, "6dof pose estimation of transparent object from a single rgb-d image," *Sensors*, vol. 20, no. 23, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/23/6790

[5] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[7] 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 *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 353–367. [Online]. Available: https://proceedings.mlr.press/v205/kerr23a.html

[8] ——, "Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects," in *6th Annual Conference on Robot Learning*, 2022.

[9] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[10] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," *CoRR*, vol. abs/2103.13744, 2021. [Online]. Available: https://arxiv.org/abs/2103.13744

[11] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *NeurIPS*, 2020.

[12] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "PlenOctrees for real-time rendering of neural radiance fields," in *ICCV*, 2021.

[13] C. Sun, M. Sun, and H. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *CVPR*, 2022.

[14] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin, "Fastnerf: High-fidelity neural rendering at 200fps," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, oct 2021, pp. 14 326–14 335. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.01408

[15] S. Lombardi, T. Simon, G. Schwartz, M. Zollhoefer, Y. Sheikh, and J. Saragih, "Mixture of volumetric primitives for efficient neural rendering," *ACM Trans. Graph.*, vol. 40, no. 4, jul 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459863

[16] K. Deng, A. Liu, J. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2022, pp. 12 872–12 881. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01254

[17] B. Attal, E. Laidlaw, A. Gokaslan, C. Kim, C. Richardt, J. Tompkin, and M. O'Toole, "Törf: Time-of-flight radiance fields for dynamic scene view synthesis," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[18] Y. Wei, S. Liu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, "Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo," in *ICCV*, 2021.

[19] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. A. Chaitanya, A. S. Kaplanyan, and M. Steinberger, "DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks," *Computer Graphics Forum*, vol. 40, no. 4, 2021. [Online]. Available: https://doi.org/10.1111/cgf.14340

[20] E. Sucar, S. Liu, J. Ortiz, and A. Davison, "iMAP: Implicit mapping and positioning in real-time," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[21] J. Tang, "Torch-ngp: a pytorch implementation of instant-ngp," 2022, https://github.com/ashawkey/torch-ngp.

[22] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-NeRF: Structured view-dependent appearance for neural radiance fields," *CVPR*, 2022.

[23] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron, "NeRF in the dark: High dynamic range view synthesis from noisy raw images," *CVPR*, 2022.

[24] J. Y. Zhang, G. Yang, S. Tulsiani, and D. Ramanan, "NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild," in *Conference on Neural Information Processing Systems*, 2021.

[25] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, "Stereo radiance fields (srf): Learning view synthesis from sparse views of novel scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2021.

[26] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, "Sparf: Neural radiance fields from sparse and noisy poses." IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023.

[27] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[28] J. Y. Zhang, A. Lin, M. Kumar, T.-H. Yang, D. Ramanan, and S. Tulsiani, "Cameras as rays: Pose estimation via ray diffusion," in *International Conference on Learning Representations (ICLR)*, 2024.

[29] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[30] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "iNeRF: Inverting neural radiance fields for pose estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[31] Y. Chen, X. Chen, X. Wang, Q. Zhang, Y. Guo, Y. Shan, and F. Wang, "Local-to-global registration for bundle-adjusting neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8264–8273.

[32] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, and J. Park, "Self-calibrating neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 5846–5854.

[33] J. Fang, T. Yi, X. Wang, L. Xie, X. Zhang, W. Liu, M. Nießner, and Q. Tian, "Fast dynamic radiance fields with time-aware neural voxels," in *SIGGRAPH Asia 2022 Conference Papers*, 2022.

[34] B. P. Duisterhof, Z. Mandi, Y. Yao, J.-W. Liu, M. Z. Shou, S. Song, and J. Ichnowski, "Md-splatting: Learning metric deformation from 4d gaussians in highly deformable scenes," 2023.

[35] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and W. Xinggang, "4d gaussian splatting for real-time dynamic scene rendering," *arXiv preprint arXiv:2310.08528*, 2023.

[36] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," in *3DV*, 2024.

[37] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," 2021.

[38] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," 2022.

[39] X. Zhang, S. Bi, K. Sunkavalli, H. Su, and Z. Xu, "Nerfusion: Fusing radiance fields for large-scale scene reconstruction," *CVPR*, 2022.

[40] E. Xie, W. Wang, W. Wang, P. Sun, H. Xu, D. Liang, and P. Luo, "Segmenting transparent objects in the wild with transformer," 08 2021, pp. 1194–1200.

[41] Y. R. Wang, Y. Zhao, H. Xu, S. Eppel, A. Aspuru-Guzik, F. Shkurti, and A. Garg, "Mvtrans: Multi-view perception of transparent objects," 2023.

[42] 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," *CoRR*, vol. abs/1703.09312, 2017. [Online]. Available: http://arxiv.org/abs/1703.09312

[43] B. O. Community, "Blender - a 3d modelling and rendering package," 2018. [Online]. Available: http://www.blender.org