NeRFuser: Scalable Scene Representation by NeRF Registration and Blending

Jiading Fang*,[1], Shengjie Lin*,[1], Igor Vasiljevic[2], Vitor Guizilini[2], Rares Ambrus[2], Adrien Gaidon[2], Gregory Shakhnarovich[1], and Matthew R. Walter[1]

[1] Toyota Technological Institute at Chicago
[2] Toyota Research Institute

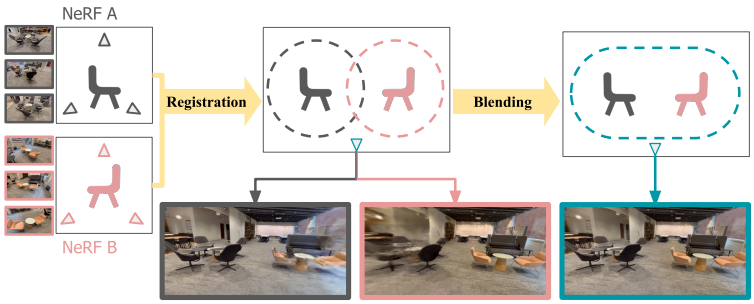


Fig. 1: Starting from separately constructed input NeRFs, NeRF A and NeRF B, we render images at novel viewpoints, including those not well-covered by either input NeRF. Our proposed **NeRFuser** method produces renderings (lower right) that are superior to those from the original NeRFs (A and B) and other baselines.

Abstract. We present **NeRFuser**, a novel framework that extends the representational capacity of neural radiance fields (NeRFs) to produce high-fidelity representations of large-scale scenes. Integral to our approach is its decomposition of spatially extended environments into a collection of small-scale scenes, each represented by an individual NeRF model (i.e., a sub-map) produced by one or more agents. Critically, we only assume access to these individual NeRFs and *not any* of the original training images or the poses with which they were trained, nor the relative transformations between the NeRFs. Towards this goal, NeRFuser adopts a two-stage procedure that consists of NeRF registration and NeRF blending. For registration, we propose *registration from re-rendering*, a technique that infers the transformation between NeRFs based on images synthesized from individual NeRFs, along with *distant accumulation*, an effective image quality measure for pose filtering. For blending, we propose *sample-based inverse distance weighting* to blend visual information at the ray sample level. We evaluate NeRFuser on an indoor dataset and two public benchmarks, showing state-of-the-art results on test views including those that are otherwise challenging to render for individual source NeRFs. Code is publicly released as a Python package at https://github.com/ripl/nerfuser.

* Equal contribution. Order is interchangeable.

1 Introduction

In order to carry out complex tasks, robots require rich representations of their environments. As such, the robotics community has paid significant attention to scene reconstruction, including the scalability to spatially extended environments [5]. Neural radiance fields (NeRFs) [43,20] provide a powerful means of generating high-fidelity, memory-efficient 3D scene representations from commodity (i.e., low-cost) cameras. In this paper, we seek to extend the representational power of NeRFs to efficiently model large-scale environments. While recent methods have shown some success at modelling large scenes with a single NeRF [22,4], its expressivity is fundamentally constrained by the model capacity. Instead, our approach achieves scalable NeRF representations by decomposing large environments into a collection of small scenes, each represented by their own individually trained NeRF. Such a decomposition not only improves quality and computational efficiency [36], but also allows mapping to be distributed among a team of robots and/or human-operated cameras.

Targeting such use cases, we propose NeRFuser (Fig. 1), a NeRF fusion framework for the registration and blending of pre-trained NeRFs. NeRFuser only requires black-box access to the constructed NeRFs, but not the original training images or the training poses, nor the relative transformation between the individual NeRFs.[3] Nonetheless, NeRFuser can register NeRFs (in terms of both pose and scale) and render images from registered NeRFs with better quality than can be achieved by individual NeRFs and other baselines. Specifically, NeRFuser fuses NeRFs in two steps: *registration* and *blending*. For the first step, we propose *registration from re-rendering*, a technique that takes advantage of NeRFs' ability to synthesize high-quality views to solve for the six-DoF pose and scale (i.e., SIM3) transformation between pairs of individual NeRFs via 2D image matching. Further, we propose *distant accumulation*, a new NeRF image quality measure for pose filtering. For the second step, we describe a fine-grained sample-based blending technique following inverse-distance-weighting (IDW).

2 Related Work

Neural Radiance Fields A NeRF [20] is an implicit representation of a 3D scene that optimizes a neural network composed of MLPs to represent the scene as density and radiance fields, which can be used to synthesize novel views through volumetric rendering. Since its introduction, many follow-up methods [2,22,3,45,34,37] have improved over the original implementation. One line of work involves the reconstruction of large-scale NeRFs [42,46,36,33,50,49,39,25,36]. However, most of these methods reconstruct the entire scene using a single model. While progressive training [42,33] and carefully designed data structures [46,49]

[3] We note that training poses are not informative for registration as they are defined with respect to individual NeRF-specific reference frames. Further, while NeRFuser can make use of training poses when available for rendering, we empirically find that images rendered at non-training poses often yield better registration results.

Fig. 2: Qualitative comparison of blending methods. Our proposed **IDW-Sample** produces high-quality blending for both chairs, while baseline methods fail on at least one chair. Notice that the blended results (e.g., IDW-Sample) may be even sharper than the ground-truth image, which exhibits motion blur.

help to expand the expressivity of a single model, other works show that a collection of many small models can perform better, while maintaining the same number of parameters [25,36,41]. Our method provides a novel way to reason over many small models, combining them to improve performance. **NeRF Registration** NeRFs are optimized from posed images, with poses usually obtained from a structure-from-motion (SfM) method [31,32,29,30,9,26,10]. Because these methods are scale-agnostic, the resulting coordinate system will have an arbitrary scale specific to each NeRF. Jointly using multiple NeRFs requires *NeRF Registration*, i.e., solving for the relative transformation between their respective coordinate systems. Note that the setting is different from "NeRF Inversion" [44,18] that estimates the six-DoF camera pose relative to the pre-trained NeRF given an image, a technique that has been used for NeRF-based localization [1,19,6]. LENS [21] proposes using NeRF to render extra images to train a per-scene pose regression network, while NeRFuser assumes neither training poses nor training images and does not require training an extra network for each scene. Also relevant are works that jointly optimize NeRF representations along with the poses and intrinsics [17,40,14]. However, NeRFuser only uses SfM on re-rendered images, and does not modify the pre-trained NeRFs themselves. Of the few works in this emerging field of NeRF registration, most [11,23,15] utilize only geometric cues—extracting surface or density fields from the NeRF representations, and thus do not take full advantage of the rich radiance information. On the other hand, DReg-NeRF [7] learns to predict 3D correspondences given the input NeRFs, thus suffering a severe performance drop for out-of-distribution scenes. Moreover, DReg-NeRF as well as nerf2nerf [11] are only capable of registering two NeRFs at a time, require known scales, and fail catastrophically for unbounded large-scale scenes. nerf2nerf [11] further requires a reasonable initialization from human annotations. In contrast, our method can register multiple

NeRFs simultaneously including scale recovery, is designed to work on large-scale real-world scenes, and does not involve human annotations.

NeRF Blending Given aligned NeRFs from registration, the next step is to merge (i.e., "blend") them. Nerflets [47] decomposes the scene with ellipsoids and blends the NeRFs according to their covariance matrices while each NeRF has access to information from all images. Blended-NeRF [12] focuses more on generative metrics rather than the reconstruction setting. More similar to our setting, Block-NeRF [36] blends NeRFs in either an image- or pixel-wise manner, similar to traditional 2D image blending techniques. In terms of blending weights, Block-NeRF uses either inverse distance weighting (IDW) or predicted visibility, where IDW is eventually adopted because of temporal consistency. Note again that we only assume access to the trained NeRFs but not the training camera poses, which are required to train the visibility network. Thus, visibility prediction is not feasible in our setting.

3D Gaussian Splatting Compared to NeRF, 3DGS [16] is a more explicit 3D representation that employs a flexible set of Gaussian blobs to encode the scene geometry and radiance field. While it is out of scope to compare 3DGS with NeRF, the need for registering and jointly rendering multiple 3DGS similarly applies. Though this work targets the use of NeRFs, some of the proposed techniques may shed inspiration on solving the same problem with this scene representation, especially for the registration.

3 Methodology

In this section, we describe our NeRF registration method "Registration from Re-rendering" and our blending technique "IDW-Sample". Without loss of generality, we consider a setting with two NeRFs A and B. The extension to three or more NeRFs is straightforward, and supported in our public code release.

3.1 Registration from NeRF Re-rendering

The first part of our framework estimates the poses of the input NeRFs. We assume that each NeRF is trained with its own set of images, and that individual NeRFs capture different, yet overlapping, areas of a larger scene. Each NeRF may have its own coordinate system (e.g., of the robot collecting the images). Our goal is to find the transformation ${}^{A}T_{B} \in \mathrm{SIM}(3)$ that transforms a 3D point p_B in NeRF B to its corresponding point p_A in NeRF A as $p_A = {}^{A}T_{B} p_B$.

Overview Since NeRFs are known for high-quality, novel view synthesis, we strategically sample a set of poses and use them as local poses to query each input NeRF to get re-rendered images. We then re-purpose off-the-shelf structure-from-motion methods [9,30] to run on the union of re-rendered images. Here, the intuition is that SfM can recover the poses of all re-rendered images in a shared coordinate system. Since the pose of each re-rendered image in its source NeRF's local coordinate system is sampled, and hence known, we can identify the transformation from the shared SfM frame to each NeRF's local frame.

Sampling Strategy of Poses for Re-rendering Though we do not assume access to the camera poses used in generating the input NeRFs, we do assume that these poses are pre-processed in a standardized way [37], i.e., that they are (i) centered so that the mean translation vector becomes the origin; (ii) rotated so that the mean up direction is aligned with z axis; and (iii) uniformly scaled so that the maximum range of poses lie along a $[-1, 1]$ axis. Accordingly, our strategy is to uniformly sample poses mainly on the upper hemisphere (due to (ii)) of radius 1 (due to (iii)) located at the origin (due to (i)). This pose sampling strategy is simple yet effective given no access to the NeRFs' training poses.

Distant Accumulation for Pose Filtering Some renderings at the sampled poses can be of poor quality nonetheless (Figure 3(a)). Although the SfM procedure is robust to poor renderings to some extent (through feature extraction and matching), it is beneficial to filter them out to reduce processing time and improve accuracy. We can use accumulation (Figure 3(b)) computed from the NeRF's volumetric rendering to measure NeRF ray uncertainty [35], since low accumulation values usually correspond to under-trained pixels. We further observe that, (i) when a ray is under-trained, the termination probability mass tends to distribute evenly; (ii) cameras inside or very close to an object usually bring poor renderings. Hence, we propose a novel ray-quality measure *Distant Accumulation*

$$q_d = \int_d^\infty T(t)\sigma(t)\mathrm{d}t, \tag{1}$$

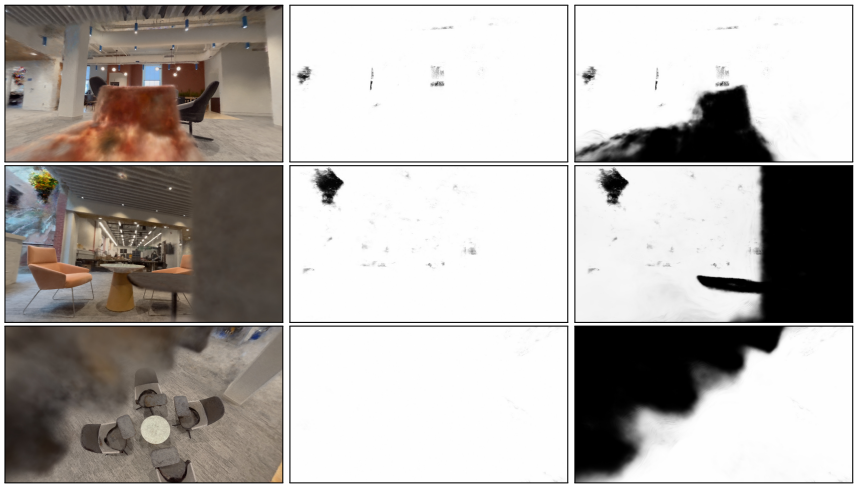
where $T(t)$ and $\sigma(t)$ denote transmittance and density, respectively, as introduced in Mildenhall et al., [20]. We use distant accumulation q_d averaged over all pixels as the image quality measure, where d is set to a moderate value within $[0, 1]$ (recall that the NeRF's coordinate system is assumed to be normalized in scale; $d = 0.3$ is used in Figure 3(c)). Intuitively, (i) when a ray is under-trained, an evenly distributed termination probability results in q_d being substantially smaller than 1.0; and (ii) when a camera is inside or too close to an object, most rays will have high termination probability at a close distance, also resulting in a small q_d. As shown in Figure 3(c), our proposed measure clearly discriminates against poor renderings. We use the mean distant accumulation to filter out poor renderings before applying the SfM procedure.

3.2 Sampled-based NeRF Blending

NeRF blending aims to integrate predictions from the input NeRFs in pursuit of high-quality novel view synthesis. There are two key questions to consider:

(i) **What to blend**: At what granularity should we blend the information?
(ii) **How to blend**: How to compute the blending weights?

Block-NeRF [36] answers question (ii) with inverse distance weighting (IDW) and predicted visibility weighting. IDW determines the contribution of each NeRF according to $w_i \propto d_i^{-\gamma}$, where d_i is some notion of distance between the center of NeRF i to the element in question (to be answered by question (i)), and



(a) RGB rendering (b) regular accumulation (c) distant accumulation

Fig. 3: Illustration of NeRF renderings and their accumulations. Column (a) shows RGB renderings. Columns (b) and (c) show the gray-scale plots for regular and distant accumulation, respectively, where white means high accumulation. The visualizations show that regular accumulations are mostly close to 1.0 regardless of the various rendering quality, while the distant accumulation clearly discriminates poorly rendered regions.

$\gamma \in \mathbb{R}^+$ modulates the blending rate. Not being the focus, the L2 distance metric is used for all IDW variants. On the other hand, visibility weighting tries to weight the element by its visibility from the camera views during NeRF training. However, a visibility prediction network is required to be trained jointly with the NeRF and used during inference, which we can not assume for typical NeRFs. Due to temporal inconsistency with visibility weighting, Block-NeRF prefers IDW weighting (specifically IDW-2D, explained below).

Block-NeRF answers question (i) with image-wise and pixel-wise blending. With image-wise blending, all pixels in an image have the same blending weights. When combined with IDW (i.e., IDW-2D), the weights are computed based on the distance between the camera center and NeRF centers. For pixel-wise blending, each pixel has its own blending weight. When combined with IDW (i.e., IDW-3D), the weights are computed from the distance between the expected point of ray termination for each pixel and NeRF centers.

In this paper, recognizing that NeRF is inherently a 3D representation with computations performed at the ray sample level, we answer question (i) by proposing a novel sample-based blending method. On the other hand, although IDW may not yield the best quality, it is nevertheless favorable for both its simplicity and being a good approximation, especially under the typical setting of surrounding camera views. Thus we still follow the IDW principle to answer question (ii) but adapt it by calculating the distance between the ray sample

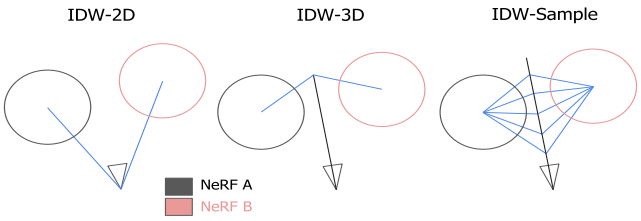


Fig. 4: Illustration of IDW-based blending methods.

and NeRF centers, achieving a finer blending result in principle. Moreover, the weighting is irrespective of depth quality, which can be an issue for NeRF and thus the major downside of IDW-3D. Since we use IDW with sample-based blending, we coin our method *IDW-Sample*. Figure 4 provides an illustration of the comparison between different methods.

Sample-wise Blending with IDW During NeRF's volumetric rendering stage, a pixel's color is computed using samples along the ray. Recognizing this, we propose a sample-wise blending method that calculates the blending weights for each ray sample using IDW. We show that the original volumetric rendering methodology can be easily extended to take advantage of these new sample-wise blending weights, resulting in our proposed *IDW-Sample* strategy.

Proximity Test NeRFs can only render with high-quality within their effective range. we introduce a *Proximity Test* first to decide which NeRFs are relevant for the given rendering pose. As we located each NeRF in a shared coordinate frame from the NeRF registration phase, we can calculate the distances between the given rendering camera position to all NeRFs and sort them in an increasing order $d_0 < d_1 < \cdots < d_{N-1}$, assuming there are N NeRFs to blend. We then divide them by the closest distance d_0 to have a set of distance ratios $(1 = \frac{d_0}{d_0} < \frac{d_1}{d_0} < \cdots < \frac{d_{N-1}}{d_0})$. A threshold τ is set to filter out NeRFs with distance ratio larger than it. Note that this threshold τ can be set in an adaptive way, for example keeping only NeRFs with k-smallest or k-th quantile τ for blending. So after the filtering, the computational complexity for NeRF blending is $\mathcal{O}(k)$.

Merging Ray Samples Consider a pixel to be rendered, which gets unprojected into a ray. Since ray samples may be separately proposed for each NeRF according to the density field at irregular intervals, we need to merge them into a single set. Given samples $\{(t_k^A, \delta_k^A)\}_k$ and $\{(t_k^B, \delta_k^B)\}_k$ proposed from NeRF A and NeRF B respectively, we merge them into a single set of ray samples $\{(\bar{t}_k, \bar{\delta}_k)\}_k$ by taking the sample location t and length δ, as illustrated in Figure 5. We update the termination probability and color of each new sample in the merged set for each source NeRF. Given a ray sample proposed by a NeRF, we assume its

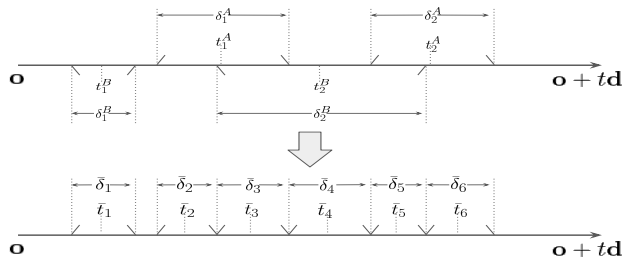


Fig. 5: An illustration of how ray samples proposed by two NeRFs are merged based on their locations and lengths. Top: two sets of ray samples proposed by NeRF A and NeRF B; Bottom: the single set of merged ray samples.

termination probability mass is uniformly distributed over its length, while its color is the same for any point within coverage. The operation is implemented to be parallel within each batch of rays to avoid being a bottleneck.

Blending Process We use IDW to compute the blending weight for each sample. Specifically, let $\boldsymbol{x}_i$ be the origin of NeRF$_i$ for $i \in \{A, B\}$, $\boldsymbol{o}$ be the camera's optical center, $\boldsymbol{r} = (\boldsymbol{o}, \boldsymbol{d})$ be the ray corresponding to pixel j to be rendered, and $(\bar{t}_k, \bar{\delta}_k)$ be a ray sample from the merged samples set, with updated termination probability mass $\bar{p}_{i,k}$ and color $\bar{\boldsymbol{c}}_{i,k}$ for each NeRF$_i$. We compute its blending weight as $w_{i,k} \propto {d_{i,k}}^{-\gamma}$, where $d_{i,k} = \|\boldsymbol{x}_i - (\boldsymbol{o} + \bar{t}_k \boldsymbol{d})\|_2$. The blended pixel j is

$$I^{(j)} = \sum_k \sum_i w_{i,k} \bar{p}_{i,k} \bar{\boldsymbol{c}}_{i,k} \tag{2}$$

Weights $w_{i,k}$ are normalized following two steps in order: (i) $\sum_i w_{i,k} = 1,\ \forall k$; and (ii) $\sum_k \sum_i w_{i,k} \bar{p}_{i,k} = 1$. Step (i) means that our method does not change the relative weighting of samples along a given ray, which is already dictated by the termination probability. Step (ii) ensures a valid color for the rendered pixel.

4 Experiments

In this section, we describe our registration and blending experiments on our self-collected object-centric indoor dataset as well as two public benchmarks: ScanNet [8] and MissionBay [36]. While our framework is agnostic to the base NeRF implementation, we use NeRFacto [37] as the base NeRF model. All NeRF training hyper-parameters follow the default settings except that we disable camera optimization to avoid effects on registration evaluation.

4.1 Datasets

Object-Centric Indoor Scenes We created a dataset consisting of three indoor scenes, using an iPhone 13 mini in video mode.[4] Each scene consists of

[4] Available at https://huggingface.co/datasets/RIPL/TTIC-common/tree/main.

Blending	Ground-truth ${}^{A}T_{B}$			Estimated ${}^{A}\hat{T}_{B}$		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF	20.92	0.716	0.369	20.90	0.714	0.370
Nearest (Block-NeRF)	23.81	0.779	0.283	23.68	0.774	0.287
IDW-2D (Block-NeRF)	24.70	0.795	0.267	24.64	0.792	0.267
IDW-3D (Block-NeRF)	23.48	0.776	0.279	23.45	0.772	0.280
IDW-Sample (Ours)	**24.91**	**0.813**	**0.228**	**24.83**	**0.810**	**0.229**

Table 1: Blending results on Object-Centric Indoor Scenes. *IDW-Sample* works the best for all metrics with both ground-truth and estimated transformations. Results with estimated ${}^{A}\hat{T}_{B}$ are only marginally worse than those with ground-truth ${}^{A}T_{B}$, which demonstrates that our proposed NeRF registration is accurate enough for the downstream blending task.

three video clips—we choose two objects in each scene, and collect two overlapping video sequences that focus on each object. We collect a third sequence that observes the entire scene as the test set. We test NeRFuser on this dataset and report results of both registration and blending.

ScanNet Dataset The ScanNet dataset provides a total of 1513 RGB-D scenes with annotated camera poses, from which we use the first 218 scenes. We downsample the frames so that roughly 200 posed RGB-D images are kept from each scene. We then split the images into three sets: two for training NeRFs and one for testing. We test NeRF registration on this dataset and compare with point-cloud registration methods.

Mission Bay Dataset To further test the rendering quality of different blending methods, we run experiments on the Mission Bay dataset from Block-NeRF [36], featuring a street scene. We report comparisons with other blending strategies.

4.2 NeRF Registration and Blending

Registration plus Blending We test NeRFuser including both NeRF registration and NeRF blending on Object-Centric Indoor Scenes. For registration, we re-render 32 poses that are roughly uniformly placed on the upper hemisphere of radius 1, with elevation from 0 to 30°. For blending, we set the distance test ratio to $\tau = 1.8$ and blending rate to $\gamma = 5$. We report numbers averaged over test images of all three scenes from our dataset in Table 1.

Registration To further test the registration performance on a large-scale dataset, we use the ScanNet dataset [8] as prepared according to Section 4.1. We repeat the same registration procedure as above, except that we sample 60 hemispheric poses. For scale error, we compute $s_{\text{err}} = |\log \Delta s|$, where Δs is the ratio of ground-truth over estimated scale. During experiments, we notice failure cases due to `NaN` or outlier values. To report more meaningful numbers, we treat cases that meet any of the following conditions as failure: (i) is `NaN`; (ii) $r_{\text{err}} > 5°$; (iii) $t_{\text{err}} > 0.2$; or (iv) $s_{\text{err}} > 0.1$. We also compare our method to various point-cloud registration (PCR) baselines using both (i) point-clouds extracted from

Registration	r_{err} (°)	t_{err}	s_{err}	Success
NeRF-extracted point-cloud				
ICP [27]	3.027	0.1151	N/A	0.13
FGR [48]	4.549	0.1844	N/A	0.04
FPFH [28]	2.805	0.0381	N/A	0.17
RGB-D-fused point-cloud				
ICP [27]	1.598	0.0816	N/A	0.17
FGR [48]	1.330	0.0372	N/A	0.71
FPFH [28]	**0.049**	**0.0205**	N/A	*0.79*
NeRFuser	*0.588*	*0.0315*	**0.0211**	**0.84**

Table 2: **Registration results on ScanNet.** We compare to point-cloud registration methods on both NeRF-extracted point-cloud and ground-truth RGB-D-fusion point-cloud. Due to the noisy geometry of NeRF reconstructions, registration performance on NeRF-extracted point-clouds is inferior. However, NeRFuser is comparable to the registration performance on RGB-D-fused methods in terms of r_{err} and t_{err}, while having the highest success rate. Unlike point-cloud baselines, our method also recovers the relative scale. **Bold** numbers are the best, *italic* numbers are second-best.

Blending	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF	17.306	0.571	0.502
Nearest	19.070	0.657	0.398
IDW-2D	19.692	0.659	0.413
IDW-3D	18.806	0.636	0.433
IDW-Sample	**19.986**	**0.678**	**0.388**

Table 3: Blending results on Mission Bay dataset.

NeRFs and (ii) point-clouds fused from ground-truth posed RGB-D images. We report in Table 2 the results of our registration method and various PCR baselines averaged over all successfully registered scenes, as well as the success rate.

Blending To further test our blending performance, we use the outdoor Mission Bay dataset as described in Section 4.1, with ground-truth transformations. We set the distance test ratio to $\tau = 1.2$, and the blending rate to $\gamma = 10$. Table 3 reports the quantitative results averaged over the test images from all scenes, while Figure 6 provides a visualization of the qualitative results.

4.3 Ablation Studies

Ablation of distant accumulation for filtering poses in NeRF registration We run NeRF registration experiments using different thresholds of the proposed mean distant accumulation for filtering poses. As shown in Figure 7, distant accumulation-based filtering removes mostly bad images when compared

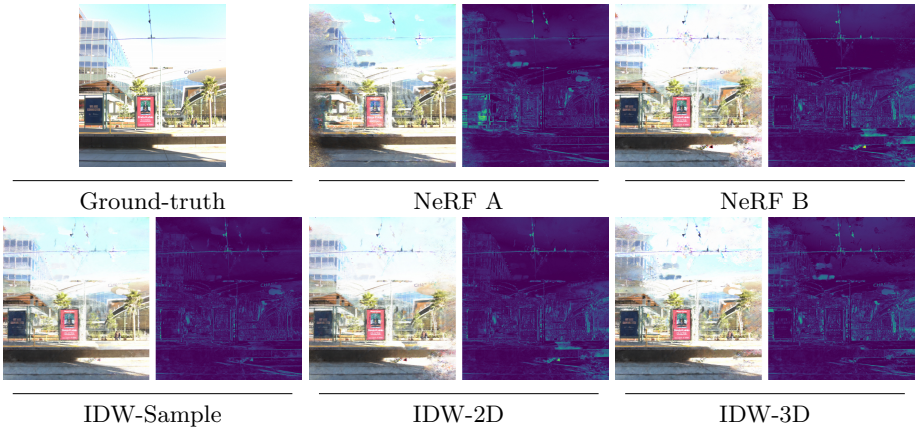


Fig. 6: NeRF blending with IDW-based methods on the Mission Bay dataset. Per-pixel errors are visualized as heat maps. Individual NeRF renderings have large artifacts on either side, which are best resolved by *IDW-Sample* blending.

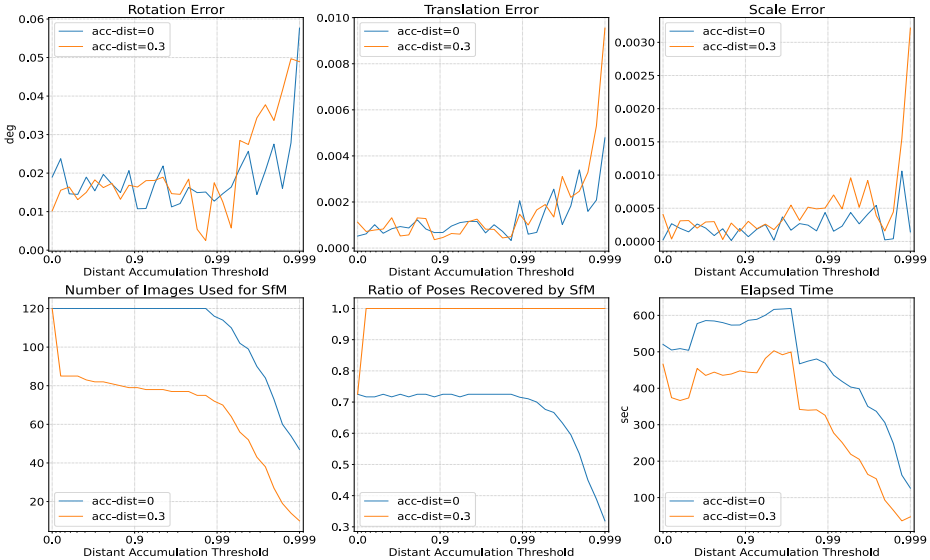


Fig. 7: Registration error and time consumption for different accumulation-based pose filtering thresholds with distant accumulation q_d ($d = 0.3$) and regular accumulation ($d = 0.0$).

to regular accumulation, leading to a significant decrease in registration time, while yielding similar registration accuracy.

Ablation on re-rendering poses for NeRF registration We study the registration performance on ScanNet dataset w.r.t. the number of sampled poses. To account for the fact that each scene may be of a different scale, we introduce ρ as the ratio of the number of sampled poses over the number of training

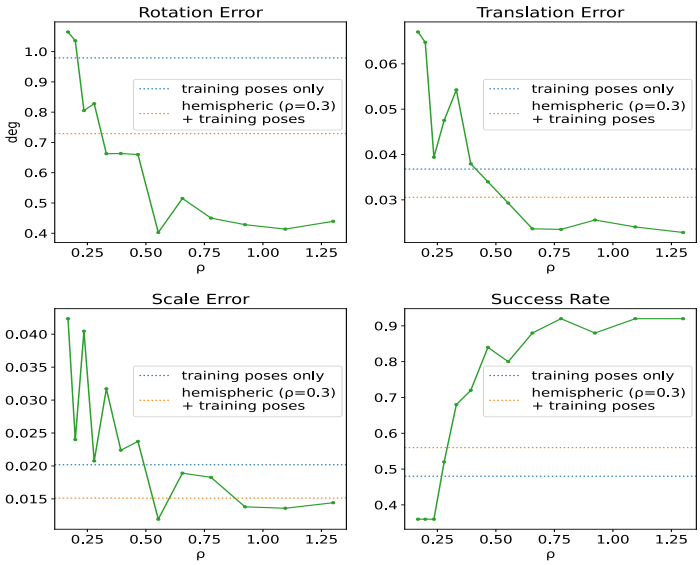


Fig. 8: Effect of re-rendering poses on NeRF registration.

views. We geometrically sample $\rho \in [0.167, 1.3]$, and generate the hemispheric poses accordingly. We evaluate the performance of NeRF registration w.r.t. ρ averaged over all ScanNet scenes. In addition, we include two more settings. (i) *training poses only*: instead of hemispheric poses, we use NeRF's training poses for re-rendering; (ii) *hemispheric + training poses*: we use NeRF's training poses together with hemispherically sampled poses ($\rho = 0.3$) for re-rendering.

Results are reported in Figure 8. With more sampled poses, the registration errors go down while success rate improves (green curve). Using additional hemispheric poses besides the training poses also proves helpful (orange line vs. blue line). Interestingly, with a large enough ratio ρ, registration with hemispherically sampled poses outperforms training poses when using the same number or fewer poses in total. This shows that it is beneficial to have a larger spatial span of re-rendering poses for registration.

Ablation of γ in IDW-based blending We study the effect of blending rate γ in IDW-based blending on Object-Centric Indoor Scenes. Specifically, we use ground-truth transformations and set the distance test ratio to $\tau = 1.8$. We geometrically sample γ in $[10^{-2}, 10^{3}]$. For each sampled γ, we blend NeRFs with all IDW-based methods and report the results averaged over test images of all three scenes from Object-Centric Indoor Scenes. We report the results in Figure 9. We draw dotted horizontal lines for *Nearest* and *NeRF*, which are not affected by γ. For all blending methods, quality initially increases with γ, but then decreases as γ increases further. As $\gamma \to 0$, all IDW-based methods become equivalent to using the mean image. As $\gamma \to \infty$, *IDW-2D* becomes the same as *Nearest*, while *IDW-Sample* becomes analogous to KiloNeRF [25]. We find

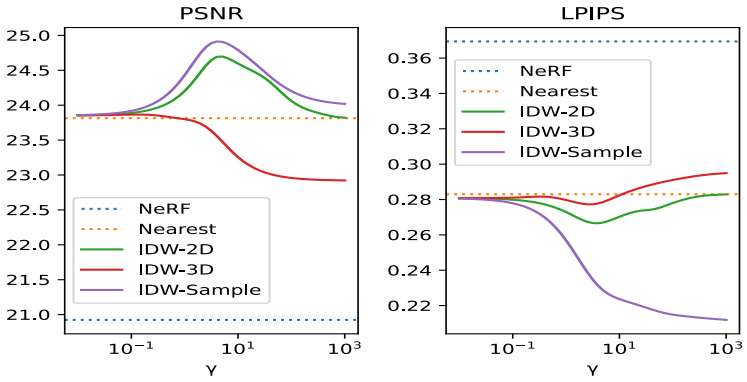


Fig. 9: Effect of blending rate γ in IDW-based blending.

the optimal γ in between the extremes for any IDW-based method. Moreover, *IDW-Sample* almost always performs the best for any given γ.

5 Limitations and Future Work

For NeRF registration, our method includes SfM and thus inherits the limitations of standard SfM pipelines, including difficulties with repetitive or flat textures and limited visual overlap. Additionally, we require NeRF scenes to be roughly static and under the same lighting conditions. To address these limitations, we are investigating improved approaches to registration that employ more robust matching algorithms [13], the use of NeRF relighting [38] to mitigate illumination changes, and dynamic NeRFs [24] to accommodate changing scenes.

For NeRF blending, a ray sample can be better trusted during blending when it receives better supervision during NeRF training. IDW-sample approximates a measure of this supervision by the distance between the ray sample and the NeRF center. However, this approximation does not take into account the ray sample's direction, and thus can be inaccurate (though we assume training poses are not available). In future work, we would like to explore new weighting methods that measure the trustworthiness of ray samples in more systematic ways.

6 Conclusion

We have introduced NeRFuser, a NeRF fusion pipeline that registers and blends arbitrary input NeRFs, extending a standard image-based processing pipeline to treat NeRFs as input data. To perform NeRF registration, we propose *registration from re-rendering*, taking advantage of NeRFs' ability to synthesize high quality novel views with *distant accumulation*, including an effective image quality measure *distant accumulation* for pose filtering. For NeRF blending, we propose *IDW-Sample*, leveraging the ray sampling nature in NeRF volumetric rendering. We believe this tool will help towards the proliferation of implicit representations as raw data for future 3D robotic applications.

References

1. Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 2021.
2. Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proc. Int'l. Conf. on Computer Vision (ICCV)*, 2021.
3. Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.
4. Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-NeRF: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023.
5. Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. on Robotics*, 2016.
6. Timothy Chen, Preston Culbertson, and Mac Schwager. CATNIPS: Collision avoidance through neural implicit probabilistic scenes. *IEEE Trans. on Robotics*, 2024.
7. Yu Chen and Gim Hee Lee. DReg-NeRF: Deep registration for neural radiance fields. In *Proc. Int'l. Conf. on Computer Vision (ICCV)*, 2023.
8. Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.
9. Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.
10. Mihai Dusmanu, Ignacio Rocco, Tomás Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint description and detection of local features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
11. Lily Goli, Daniel Rebain, Sara Sabour, Animesh Garg, and Andrea Tagliasacchi. nerf2nerf: Pairwise registration of neural radiance fields. *arXiv preprint arXiv:2211.01600*, 2022.
12. Ori Gordon, Omri Avrahami, and Dani Lischinski. Blended-NeRF: Zero-shot object generation and blending in existing neural radiance fields. *arXiv preprint arXiv:2306.12760*, 2023.
13. Xingyi He He, Jiaming Sun, Yifan Wang, Sida Peng, Qi-Xing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. *arXiv preprint arXiv:2306.15669*, 2023.
14. Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *Proc. Int'l. Conf. on Computer Vision (ICCV)*, 2021.
15. Han Jiang, Ruoxuan Li, H.K. Sun, Yu-Wing Tai, and Chi-Keung Tang. Registering neural radiance fields as 3D density images. *arXiv preprint arXiv:2305.12843*, 2023.
16. Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. on Graphics*, 2023.

17. Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. In *Proc. Int'l. Conf. on Computer Vision (ICCV)*, 2021.
18. Yunzhi Lin, Thomas Müller, Jonathan Tremblay, Bowen Wen, Stephen Tyree, Alex Evans, Patricio A Vela, and Stan Birchfield. Parallel inversion of neural radiance fields for robust pose estimation. *arXiv preprint arXiv:2210.10108*, 2022.
19. Dominic Maggio, Marcus Abate, J. Shi, Courtney Mario, and Luca Carlone. Loc-NeRF: Monte Carlo localization using neural radiance fields. *arXiv preprint arXiv:2209.09050*, 2022.
20. Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. European Conf. on Computer Vision (ECCV)*, 2020.
21. Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. LENS: Localization enhanced by NeRF synthesis. In *Proc. Conf. on Robot Learning (CoRL)*, 2022.
22. Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics*, 2022.
23. Casey Peat, Oliver Batchelor, Richard Green, and James Atlas. Zero NeRF: Registration with zero overlap. *arXiv preprint arXiv:2211.12544*, 2022.
24. Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.
25. Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs. In *Proc. Int'l. Conf. on Computer Vision (ICCV)*, 2021.
26. Jérôme Revaud, Philippe Weinzaepfel, César Roberto de Souza, No'e Pion, Gabriela Csurka, Yohann Cabon, and M. Humenberger. R2D2: Repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019.
27. Szymon M. Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Proc. Int'l Conf. on 3-D Digital Imaging and Modeling*, 2001.
28. Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2009.
29. Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
30. Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.
31. Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.
32. Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. European Conf. on Computer Vision (ECCV)*, 2016.
33. Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. iMAP: Implicit mapping and positioning in real-time. In *Proc. Int'l. Conf. on Computer Vision (ICCV)*, 2021.
34. Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

35. Niko Sünderhauf, Jad Abou-Chakra, and Dimity Miller. Density-aware NeRF ensembles: Quantifying predictive uncertainty in neural radiance fields. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2022.
36. Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
37. Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *Proc. ACM SIGGRAPH Conf.*, 2023.
38. Marco Toschi, Riccardo De Matteo, Riccardo Spezialetti, Daniele De Gregorio, Luigi Di Stefano, and Samuele Salti. ReLight my NeRF: A dataset for novel view synthesis and relighting of real world objects. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.
39. Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-NeRF: Scalable construction of large-scale NeRFs for virtual fly-throughs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.
40. Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.
41. Xiuchao Wu, Jiamin Xu, Zihan Zhu, Hujun Bao, Qixing Huang, James Tompkin, and Weiwei Xu. Scalable neural indoor scene rendering. *ACM Trans. on Graphics*, 2022.
42. Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. CityNeRF: Building NeRF at city scale. *arXiv preprint arXiv:2112.05504*, 2021.
43. Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022.
44. Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. iMeRR: Inverting neural radiance fields for pose estimation. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2021.
45. Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.
46. Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. NeRFusion: Fusing radiance fields for large-scale scene reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
47. Xiaoshuai Zhang, Abhijit Kundu, Thomas A. Funkhouser, Leonidas J. Guibas, Hao Su, and Kyle Genova. Nerflets: Local radiance fields for efficient structure-aware 3D scene representation from 2D supervision. *arXiv preprint arXiv:2303.03361*, 2023.
48. Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Proc. European Conf. on Computer Vision (ECCV)*, 2016.
49. Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. NICER-SLAM: Neural implicit scene encoding for RGB SLAM. In *Proc. Int'l. Conf. on 3D Vision (3DV)*, March 2024.
50. Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. NICE-SLAM: Neural implicit scalable encoding for SLAM. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.