

VF-NeRF: Viewshed Fields for Rigid NeRF Registration

Leo Segre¹ and Shai Avidan¹

Tel Aviv University

https://leosegre.github.io/VF_NeRF/

Abstract. 3D scene registration is a fundamental problem in computer vision that seeks the best 6-DoF alignment between two scenes. This problem was extensively investigated in the case of point clouds and meshes, but there has been relatively limited work regarding Neural Radiance Fields (NeRF). In this paper, we consider the problem of rigid registration between two NeRFs when the position of the original cameras is not given. Our key novelty is the introduction of Viewshed Fields (VF), an implicit function that determines, for each 3D point, how likely it is to be viewed by the original cameras. We demonstrate how VF can help in the various stages of NeRF registration, with an extensive evaluation showing that VF-NeRF achieves SOTA results on various datasets with different capturing approaches such as LLFF and Objaverse. Our code will be made publicly available.

Keywords: Neural radiance fields · 3D registration · Normalizing-flows

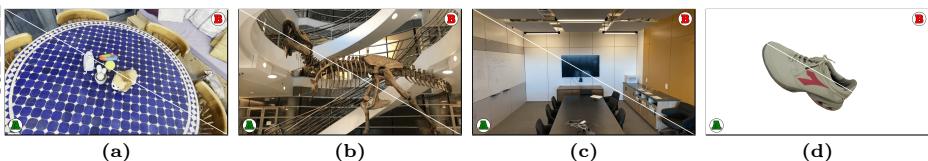


Fig. 1: NeRF Registration: Registration results of VF-NeRF for four different scenes. All images are from novel view points where the bottom left part is taken from one NeRF and the top right part is taken from the registered second NeRF. **1a** is of a casually captured scene, **1b** and **1c** are taken from LLFF dataset and **1d** is from the Objaverse dataset.

1 Introduction

Registering two 3D scenes is a fundamental problem that has been studied for years in the field of computer vision. Solving it has a large number of applications in many fields ranging from pure image processing such as medical imaging [39] and object detection [5] to world-scale tasks such as mobile robotics [33] and autonomous driving [49]. Until recently, common ways to represent, and register,

3D scenes were based on point clouds or meshes. Recently, Neural Radiance Fields (NeRF [30]) emerged as a viable alternative, and we propose a registration algorithm that operates directly on them.

Our approach is simple and straightforward. Generate a set of images from the source NeRF and seek a rigid transformation that minimizes a photometric loss of these images with respect to the target NeRF. The transformation that minimizes this loss is the one that registers the two NeRFs. Figure 1 shows registration results from several different scenes.

Given only two NeRFs, without the position of the original cameras in either one of them, we are faced with the following question: how to sample "good" virtual camera viewpoints? A possible solution is to sample the position of the virtual camera to be on the unit sphere and point the camera at the origin of the scene. However, as shown in Figure 2, this often leads to poor results, where almost half of the image is essentially noise (red-marked image on the left). Instead of working at the camera level, we aggregate the information of all the original cameras into a novel representation, termed Viewshed Fields (VF), that allows us to generate images like the two green-marked images shown on the right.

VF is an implicit function, similar to NeRF, that, given an oriented point (*i.e.*, a 3D point and a viewing direction), outputs a scalar that represents how well was the 3D point covered, from a specific direction, by the original set of images that was used to create the NeRF. If we had access to such an oriented point, we could have placed a virtual camera that is looking at it. Unfortunately, VF is an implicit function, and we do not have such access. To overcome this challenge, we treat the problem as a generative process, where the goal is to generate high VF score points. Specifically, we use Normalizing Flows (NF) [12] to map high value VF points to a Gaussian distribution in latent space during the original NeRF training. Then, we use the generative process to generate (*i.e.*, sample) high value VF points and direct the virtual camera at them. An overview of our method is shown in Figure 3.

We also use VF to help initialize the registration process and to help optimization. Specifically, we can use VF to generate a 3D point cloud and rely on the vast literature of point registration to obtain a good initial alignment. Then, during photometric optimization we rely on VF to select high quality rays that lead to better optimization results. To summarize, we make the following contributions:

- We introduce a novel representation, termed Viewshed Fields (VF), that helps register two NeRFs. VF represents 3D points that were well covered by the initial set of cameras that captured the scene.
- We generate meaningful novel views to support NeRF registration task. We use a generative method, based on Normalizing Flows, to generate high score VF points. These points are then used to set the parameters of virtual cameras which, in turn, produce images of the scene that are used to solve the NeRF registration problem.
- We show how to use VF to construct point clouds of a scene.

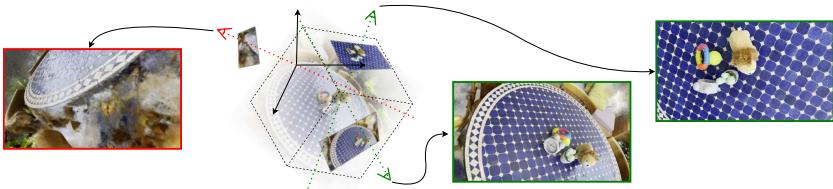


Fig. 2: Novel View Generation: Randomly sampling novel camera parameters often lead to non-informative images. For example, the red-marked image on the left was generated using a camera that lies on the unit sphere looking towards the origin. In contrast, using our novel Viewshed Fields (VF) representation we are able to generate informative camera positions (green marked images on the right) that can then be used to register two NeRFs.

2 Related Work

2.1 Neural Radiance Fields

Common ways to represent 3D scenes include point clouds, meshes, and voxel grids. Recently, NeRF [30] became a popular choice for representing 3D scenes. It introduced a differentiable renderer to optimize the 3D scene based on 2D RGB images. Utilizing the differentiable renderer, the 3D scene can be learned through back-propagation, hence the scene can be represented by the weights of a neural network, mainly MLP-based. Numerous works leveraged this approach to improve the scene quality [2, 3], accelerate the learning and rendering time [17, 31], utilize depth supervision [11] and render more complex fields such as features and semantics [16, 43, 44, 50].

A large body of work considered the use of NeRF in dynamic setting [25]. This typically requires the alignment of multiple frames to a canonical coordinate system. This alignment is usually achieved in the form of flow estimation and *not* through the estimation of a rigid global transformation, as is done here. Recently, Gaussian Splatting [21] was introduced as a powerful alternative to NeRF. However, this differs from our work since Gaussians are explicit models as opposed to the implicit nature of NeRFs.

2.2 3D Scene Registration

Registration is a well-studied task in the field of computer vision, with a wide range of works related to explicit representations such as point clouds or meshes. Due to the explicit nature of those representations, registration algorithms can work on a finite set of points or vertices and fit the two sets using classic algorithms [1, 4]. Some iterative algorithms use local refinements to tackle the registration task, such as ICP variants [8, 24, 34, 52], but those algorithms are prone to fail given a bad initialization or partial overlap between the sets. In addition to those, there are classic global methods that first match pairs between the two sets [14, 19, 35, 42] and then use a sparse subset of them for global

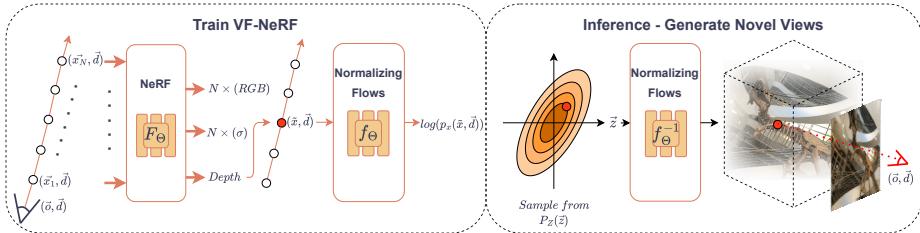


Fig. 3: VF-NeRF: (Left) Our VF-NeRF consists of two parts. The first is a NeRF network with the standard RGB and σ outputs with depth estimation. The second part is a simple normalizing-flows network, where its input is a point on the surface and the camera direction (i.e., an *oriented point*) and its output is the log-likelihood estimation that is maximized during the training phase. (Right) To generate novel views we sample from the 6-dimensional Gaussian in the Normalizing-Flows latent space. Then we recover the oriented point (\tilde{x}, \tilde{d}) and use equation 5 to reconstruct the camera origin. Finally, we render the view of the camera in position \mathbf{o} and direction \mathbf{d} .

alignment [51]. Recent works learn the alignment features utilizing deep neural networks [9, 20, 45, 48]. One method that utilize it for NeRF registration is DReg-NeRF [6] that converts the NeRF to voxel grid and train a deep neural network for registration task. It achieves improved results over point cloud registration methods but requires a large training set.

NeRF assumes that camera pose is recovered through Bundle Adjustment, such as COLMAP [36], in a pre-processing step. Recent works such as BARF [26] or L2G-NeRF [7] demonstrate that bundle adjustment can be done during NeRF training over photometric loss. This line of work takes images as input and outputs a single consistent NeRF. It does not register two NeRFs, as we do. A work that is closer to us is that of iNeRF [46]. They do not perform NeRF registration, but they do perform image to NeRF registration which can be used for NeRF registration. Their method is based on back-propagating the photometric loss through the NeRF weights to optimize the camera pose. NeRF2NeRF [18] is another method that works directly with NeRF representation. They show an improvement over point cloud registration methods, but require user input at the initialization, which is not needed in our approach.

Our work is not to be confused with implicit Signed Distance Function works [8, 32] that use a latent code-conditioned feed-forward decoder network, or directly fit a Neural Network to a point cloud to generate the surface of a shape. Our goal is not to reconstruct the shape of surfaces in the scene, but rather generate "good" view points.

2.3 Normalizing Flows

Normalizing flows (NF) is a family of invertible models that convert data from a "real-world" distribution to a standard distribution latent space and vice versa. That attribute makes NF a generative model that makes it possible to recon-

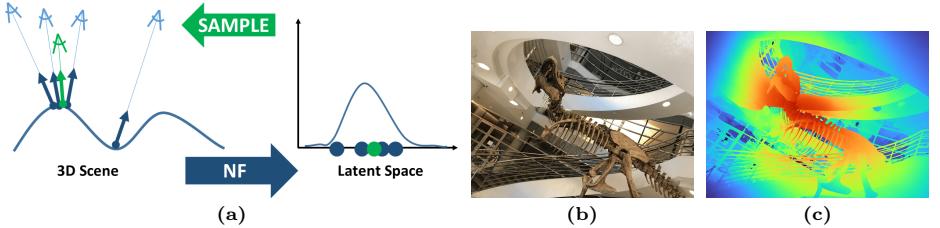


Fig. 4: Viewshed Fields: (4a Left) During NeRF training, we sample *oriented points* (blue) around surfaces in the scene and use Normalizing Flows (NF) to map them to a Gaussian in latent space. (4a Right) During NeRF registration, we sample a high visibility oriented point (green) from the Gaussian and map it to the input space where it is used to determine the position of the novel camera. 4b demonstrates a novel view synthesis generated using our method and 4c is the viewshed map generated respectively.

struct data from the original distribution by sampling the known distribution in latent space.

Real-NVP [13] proposed an invertible network that is based on MLP and affine coupling layer. The training process maximizes the log-likelihood using a change of variables. This method performs well on low dimensional data, but struggles in the case of high dimensional data such as images and videos. Moreover, the computational effort is proportional to the dimension of the data. Glow [23] proposed using invertible 1×1 convolution and a multi-scale architecture to deal with the input dimensionality. Note that in our case, we adopt Real-NVP due to the low dimensionality of our input (6D oriented points).

There has been work on trying to measure uncertainty in NeRF representations. For example, NeRF-W [28] decompose the scene into static and transient objects (i.e., walking people in static scenes). Other works presented uncertainty as an evaluation metric for novel views [38, 40]. These metrics can assist in understanding whether a novel view quality is good or not, but it can not assist in locating the scene and generate high-quality novel views. Recently, Conditional-Flow NeRF [37] introduced an NF-based method to measure uncertainty in NeRF. Although this method uses NF, it maximizes the pixel-color likelihood as a function of the NeRF outputs. Hence, the invertible nature of NF is ineffective when trying to locate the scene and generate a forward-facing novel view.

3 Method

We wish to find a 6-DoF transformation between two NeRFs. We do that by generating images from one NeRF and seeking the 6-DoF transformation that will minimize a loss between the images from one NeRF with respect to the other NeRF. Our solution is based on a novel representation, termed Viewshed Fields (VF). See Figure 4.

VF works as follows. During NeRF training we obtain *oriented points*, which are 3D surface points accompanied by a viewing direction (denoted by blue circles with pointing arrow on the left side of Figure 4a). The oriented points are mapped to a Gaussian in latent space using Normalizing Flows (the right part of Figure 4a). Once training is done, we can sample the Gaussian in latent space to generate high visibility points (marked in green). These points are points on the surface of objects in the scene that are, with high probability, observed by many cameras. Given the sampled oriented point, we can generate the position and orientation of the novel camera (marked with green in the left part of Figure 4a). Figures 4b and 4c show an example novel view, and its corresponding viewshed map, that were generated by sampling VF.

3.1 Viewshed Fields

The Viewshed Field is learned during the training phase of VF-NeRF, either together with the density and RGB values, or after the NeRF parameters are fixed (*i.e.* working on a pre-trained NeRF). We use Normalizing-Flows [12] to learn it, where the normalizing-flows model $f : \mathcal{X} \rightarrow \mathcal{Z}$, learns a mapping between the data distribution \mathcal{X} of oriented points (\mathbf{x}, \mathbf{d}) with location \mathbf{x} and direction \mathbf{d} , to a 6 dimensional diagonal Gaussian in latent space \mathcal{Z} . We learn the mapping f using Real-NVP [13] architecture with 4 layers. The optimization is done by minimizing the unsupervised negative log-likelihood of $p_X(x)$ through the typical change of variables formula of normalizing flows:

$$p_x(x) = p_z(f(x)) \cdot \left| \det \left(\frac{df}{dx} \right) \right| \quad (1)$$

$$\log(p_x(x)) = \log(p_z(f(x))) + \sum_{i=1}^K \log \left| \det \left(\frac{df_i}{df_{i-1}} \right) \right| \quad (2)$$

The data (\mathbf{x}, \mathbf{d}) is sampled during NeRF training, where \mathbf{d} is constant along a ray and \mathbf{x} varies along it. We sample only the surface of the object along the ray into the Normalizing-flows model. That is, let $\tilde{\mathbf{x}}$ be the single \mathbf{x} that lies on the surface of the object and let \mathbf{o} be the origin of the ray. The depth is simply defined by the median of the weights (of the densities learned by NeRF) accumulated along the ray.

$$\tilde{\mathbf{x}} = \mathbf{o} + \text{depth} \cdot \mathbf{d} \quad (3)$$

Now we can sample the pair $(\tilde{\mathbf{x}}, \mathbf{d})$ into the normalizing-flows model so the viewshed score (*i.e.* likelihood of the VF according to the Gaussian) is high on the surface of an object but low everywhere else, as can be seen in figure 4c.

Novel Views Since we have used Normalizing Flows to learn the Viewshed Field, we can sample points from the 6-dimensional Gaussian and map them to oriented points with high viewshed value.

$$(\mathbf{x}, \mathbf{d}) = f^{-1}(\mathbf{z}) \quad (4)$$

These oriented points' log probability is formally given by Equation 2, where $f(x) = z$. In practice, we generate K samples from \mathcal{Z} , invert them to K oriented points (\mathbf{x}, \mathbf{d}) , and finally select the top- N in terms of $\log(p_x(x))$. From each oriented point (\mathbf{x}, \mathbf{d}) we can generate a novel view since we know, with high probability, that it is valid to view the point \mathbf{x} from direction \mathbf{d} . Setting the camera origin is done by inverting equation 3.

$$\mathbf{o} = \mathbf{x} - \text{depth} \cdot \mathbf{d} \quad (5)$$

Finally, we render a novel view from the camera with origin \mathbf{o} that is facing toward the direction \mathbf{d} . High viewshed score is not guaranteed for all the pixels in this novel view, so we render the viewshed field and use a threshold to generate a 2D viewshed mask and sample accordingly, as can be seen in Figure 5.

3.2 \hat{T}_0 Initialization

Let $T \in SE(3)$ be the transformation from scene A to scene B . Our approach is based on finding T that minimizes a photometric loss, hence the initialization of T is important. We introduce two different approaches for initialization. Both are based on VF.

Photometric based initialization: We use VF_A of scene A to generate a set $C_A = \{C_i | i = 1, 2, 3...N\}$ of *good* camera view points of scene A . Given a possible transformation $T \in SE(3)$, we use VF_B of scene B to determine how well the cameras in C_A , that are transformed by T , observe *good* points in scene B . This intuition is captured in the following:

$$\text{Score}_{init}(C_A; T) = \text{Median}_{C_A}(\sum_{p \in T(C_i)} VF_B(p)) \quad (6)$$

Where $p \in T(C_i)$ refers to all oriented points p that come from camera C_i that underwent transformation T . $VF_B(p)$ is the likelihood of the oriented point p , according to Normalizing Flows, in scene B . For robustness, we take the score of T to be the median over all transformed cameras in the set C_A . For photometric initialization, we sample, at random, multiple transformations T , and pick the one with the highest score.

VF-based point cloud: The second approach we use is based on converting NeRF to point cloud and relying on the vast literature for point cloud registration. We use VF to sample the point clouds. Specifically, we sample M points from the 6-dimensional Gaussian which is the NF latent distribution and use equation 4 to reconstruct M oriented points. Then, we sample NeRF with the oriented points as its input to get the corresponding density and RGB. Finally, we utilize density values along with a specified threshold to filter out uncertain points. We currently do not integrate the colors for registration purposes. Once the point clouds of both scenes are generated, we can use any known global registration for point clouds, to serve as our initial guess.



Fig. 5: VF pixel sampling: (Left) VF map of a novel view from the Fern scene from LLFF dataset (Middle) Green pixels sampled using VF map (right) Red pixels sampled randomly. The VF mask guides the process to sample pixels with more reliable RGB value.

3.3 Gradient Based Optimization

Our algorithm performs a gradient-descent optimization. Given a set of novel views, C_A , from scene A and their corresponding viewshed maps, for each iteration i sample a set of n rays R_i with high VF score as shown in figure 5. Then utilize R_i and the fixed NeRF parameters of scene B (i.e. Θ_b) to optimize the 6-DoF parameters. \hat{T}_i is the estimated transformation at step i .

$$\mathcal{L}_{\text{photo}}(\hat{T}_{i-1}|R_i, \Theta_b) = \frac{1}{n} \sum_{r \in R_i} \left(I(r) - I(\hat{T}_{i-1}(r)) \right)^2. \quad (7)$$

The term $I(r)$ and $I(\hat{T}_i(r))$ denotes the pixel value generated from the ray r before and after the transformation \hat{T}_i applied.

We follow iNeRF [46] Gradient-Based SE(3) Optimization with the following changes. First, the optimization is done over all images in the set C_A instead of a single image. In other words, the rays in each batch are sampled from images across the set. This approach makes it easier to cover the scene geometry and it improves the results as can be seen in Table 4 (a). Second, we use the thresholded viewshed images to select only rays with high confidence. It is important to sample only where the RGB values are known with high probability, since NeRF might have artifacts outside the region of interest. Table 4 (a) shows the importance of this masking approach.

We use SGD optimizer as we found it more reliable when T_0 is not precise. At the end of this optimization, we get the $T \in SE(3)$ that minimizes the photometric loss (Eq. 7). This is the rotation and translation that represent the registration between scene A and scene B according to our approach.

4 Experiments

We evaluate our method on three different datasets: the standard LLFF [29] dataset, which consists of forward-facing camera poses across a plane. A small dataset of two casually collected videos captured by us that focus on object-centric scenes, and finally, the recently introduced synthetic Objaverse [10] dataset. See supplementary material for additional results and experiments.

Table 1: LLFF Results: Rotation and translation RMS error comparison on LLFF dataset, divided into the three experiment types regarding overlap between the frames. The results are the mean error over all the scenes in the dataset, Δt denotes the RMS translation errors multiplied by 1e2. FPFH + RANSAC uses point clouds generated by Viewshed Fields as suggested in section 3.2. VF-NeRF + PC Init refers to our algorithm after initializing with VF-based points clouds using FPFH + RANSAC. VF-NeRF + Photo Init refers to our method using photometric based initialization.

Model	Full Overlap		Partial Overlap		No Overlap	
	$\Delta t \downarrow$	$\Delta R \downarrow$	$\Delta t \downarrow$	$\Delta R \downarrow$	$\Delta t \downarrow$	$\Delta R \downarrow$
FPFH [35] + RANSAC [15]	2.1149	1.5316	3.6646	2.7126	2.2679	2.8128
iNeRF [46]	22.8907	16.1153	20.3248	12.5299	8.9063	10.0933
iNeRF [46] + Photo Init.	0.2254	0.1624	13.0270	2.5051	2.8350	5.6513
VF-NeRF + Photo Init.	0.0151	0.0206	0.0393	0.0358	0.0324	0.0358
VF-NeRF + PC Init.	0.0162	0.0157	0.0357	0.0345	0.0249	0.0286

4.1 Real World Datasets

Setting: In the case of the LLFF and casual datasets, we run COLMAP [36] on the entire set of images for each scene. Then, we split the frames into two sets that serve as the input to NeRF A and NeRF B, respectively. In particular, we evaluate three levels of overlap between the two sets of images. The simplest scenario is "full overlap", where NeRF A gets the even frames and NeRF B gets the odd frames. Another scenario is "partial overlap", where NeRF A gets the first 70% even frames and NeRF B gets the last 70% odd frames, resulting in a 40% overlap. The last scenario is "No overlap", where NeRF A gets the first half of frames and NeRF B gets the second half of frames.

To set the ground truth, we draw, for each experiment, a transformation $T \in SE(3)$ by randomizing the 6-DoF parameters, three rotation parameters within $[0, 45^\circ]$, and three translation parameters within $[-0.25, 0.25]$. Next we apply transformation T^{-1} to the camera poses of the set of images of A, and then train VF-NeRF for both sets of images. Our goal is to find the transformation $\hat{T} \in SE(3)$ that is the closest to T in terms of rotation and translation. It should be noted that COLMAP minimizes a *geometric* error while we (as well as iNeRF for that matter) minimize a *photometric* loss. Ideally, the two should coincide but this is not always the case. We extensively discuss this in the supplemental.

We evaluated the VF-based point cloud experiments once, but due to the stochastic nature of random initialization, we evaluated all other experiments 10 times and chose the result with the highest PSNR.

LLFF Dataset: Local Light Field Fusion (LLFF) [29] is a widely used dataset that includes real-world complex scenes. Specifically, we evaluated VF-NeRF on the 4 common scenes - fern, trex, horns, and room. Each scene is captured from a single plane and consists of 20-62 images, depending on the scene. NeRF is very sensitive to the view direction so generating novel views in this dataset must be as near as possible to the original captured view plane.

Table 2: Casually Captured Results: Rotation and translation RMS error comparison on two Casually Captured Real-World scenes with partial overlap between the frames. Δt denotes the RMS translation errors multiplied by $1e2$. FPFH + RANSAC uses point clouds generated by Viewshed Fields as suggested in section 3.2. VF-NeRF + PC Init refers to our algorithm after initializing with VF-based points clouds using FPFH + RANSAC. VF-NeRF + Photo Init refers to our method using photometric based initialization.

Model	Lion		Table		Average	
	$\Delta t \downarrow$	$\Delta R \downarrow$	$\Delta t \downarrow$	$\Delta R \downarrow$	$\Delta t \downarrow$	$\Delta R \downarrow$
FPFH [35] + RANSAC [15]	2.9714	2.9534	1.0842	1.2480	2.0278	2.1007
iNeRF [46]	55.3138	17.3639	8.7851	16.4886	32.0495	16.9262
iNeRF [46] + Photo Init.	0.0639	0.0238	0.0382	0.0591	0.0510	0.0414
VF-NeRF + Photo Init.	0.0157	0.0384	0.0195	0.0317	0.0176	0.0351
VF-NeRF + PC Init.	0.0292	0.0380	0.0125	0.0227	0.0209	0.0303

Our results on the LLFF dataset are reported in Table 1. It compares a variety of methods. FPFH [35] + RANSAC [15] is a point cloud based registration method. The point clouds themselves are generated using Viewshed Fields. iNeRF [46], without and with initialization, minimizes a photometric loss, like we do. Then, we evaluate our VF-NeRF with photometric based, as well as point-cloud based initializations.

As can be seen, point-cloud registration FPFH+RANSAC (first row in the table) converges but is not very accurate. iNeRF by itself does not converge (second row of the table). It does converge in case our VF-based photometric initialization is being used (third row of the table). Finally, our method VF-NeRF converges to the most accurate solution with both types of initialization (photometric or point-cloud based). Errors below 0.05 are hardly noticeable visually. In particular, the last row of the table shows that VF-NeRF improves the initial guess provided by FPFH+RANSAC (i.e., first row of the table) by two orders of magnitude. VF-NeRF dominates all other methods in all overlap scenarios.

Casually Captured Scenes One of the advantages of NeRF is the simplicity of capturing a scene in-the-wild and reconstructing the 3D model directly from the video. To evaluate the capabilities of NeRF registration on such scenes, we captured two 360° scenes using a camera phone. Each scene includes 300-400 images and an example of generated point clouds of the scenes can be seen in figure 6, more images from the dataset appear in the supplementary. We repeat the same experiment we did with the LLFF dataset and compare with the same methods. The results are reported in Table 2. As can be seen, results are very similar to the LLFF experiment. VF-NeRF (with either one of the possible initializations) dominates other methods.

Table 3: Objaverse Results: Quantitative results of registration, organized by mean value of the best relative rotation errors $\Delta\mathbf{R}$. For example, the column titled 50% we sort all scenes in ascending order of their registration rotation error and compute the mean error of the best 50% scenes. $\Delta\mathbf{t}$ denotes the relative translation errors multiplied by 1e2 with unknown scales. DReg_{df} refers to DReg with density fields and DReg refers to DReg with surface field. FPFH + RANSAC uses point clouds generated by Viewshed Fields as suggested in section 3.2. VF-NeRF + PC Init refers to our algorithm after initializing with VF-based points clouds using FPFH + RANSAC. The results of FGR, REGTR and DReg are taken from [6].

Model	50%		75%		90%		100%	
	$\Delta\mathbf{t}$	$\Delta\mathbf{R}$	$\Delta\mathbf{t}$	$\Delta\mathbf{R}$	$\Delta\mathbf{t}$	$\Delta\mathbf{R}$	$\Delta\mathbf{t}$	$\Delta\mathbf{R}$
FGR [51]	5.33	13.20	8.23	16.98	12.79	46.85	13.90	61.59
REGTR [47]	35.08	65.98	42.87	93.84	43.11	105.58	43.31	113.78
DReg _{df} [6]	5.43	18.62	12.17	56.82	14.45	74.32	16.06	86.23
DReg [6]	3.24	5.33	3.61	7.38	3.77	8.59	3.85	9.65
FPFH [35] + RANSAC [15]	1.94	1.96	1.82	2.79	2.19	3.69	3.01	9.75
VF-NeRF + PC Init.	0.47	0.26	0.57	0.60	1.08	1.11	2.14	6.77

4.2 Objaverse Dataset

Objaverse [10] is a large dataset that contains more than 800K 3D objects. DReg [6] utilized this dataset to construct a dataset of 1700+ training scenes and 44 evaluation scenes for the task of NeRF registration. Since our method is optimization based, the training set is not required and we evaluated our method directly on the 44 evaluation scenes. To initialize our method we first used VF to sample oriented points, choose points with density larger than 10 and reconstructed a point cloud for each scene. Then we used FPFH [35] + RANSAC [15] to find an initial alignment between the two point clouds.

We compare VF-NeRF to several point-cloud based registration methods, including FGR [51], REGTR [47], FPFH [35]+RANSAC [15], and two flavours of DREG [6]. We report results in Table 3. The table consists of multiple columns, that correspond to different fractions of the dataset. For example, in the column titled 50% we sorted all 44 scenes in ascending order, according to their rotation error, took the 22 scenes with the lowest error and computed their mean error. As can be seen, we dominate the table on both rotation and translation errors. Specifically, our method shows a notable improvement over FPFH [35] + RANSAC [15] (that serves as initialization to our VF-NeRF method).

We conclude that VF-NeRF performs well across all types of scenes both real and synthetic, and that VF can be used to help initialize the optimization.

4.3 Ablation Study

We conducted a number of ablation studies to evaluate different aspects of our approach.

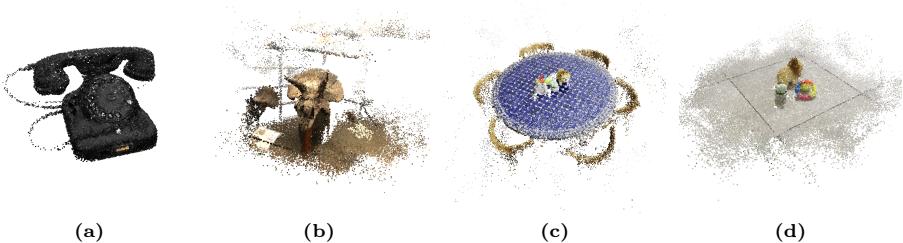


Fig. 6: Point clouds from VF: Point clouds generated by sampling from the VF distribution as explained in sub section 3.2. Each point cloud here is a combination of two point clouds from two NeRFs after applying our registration method. The examples are taken from all the datasets we evaluate in the paper, 6a from Objaverse dataset, 6b from LLFF dataset and 6c-6d from our casually captured dataset.

Table 4: (a) Ablation study: impact of each component of our method on performance. We report mean RMS error of rotation and translation over all scenes in the LLFF dataset. Δt denotes the RMS translation errors multiplied by $1e2$ (b) Noise robustness study: Impact of noise on estimation the position of the oriented points. Results are reported on the Trex scene from the LLFF dataset. The noise column denotes the amount of uniform noise (as percentage of scene size) that is added. The results are the mean error over the three overlap settings. As can be seen, VF-NeRF is robust to this noise.

Model	$\Delta t \downarrow$	$\Delta R \downarrow$
No initialization	39.2993	21.5474
No VF Masks	1.7348	1.3609
Single image	9.4008	8.9252
Full method	0.0151	0.0206

(a) Ablation Study

Noise	$\Delta t \downarrow$	$\Delta R \downarrow$
0%	0.0113	0.0108
1%	0.0114	0.0120
5%	0.0117	0.0106
10%	0.0265	0.0140
20%	0.0149	0.0155

(b) Noise robustness study

VF Abalation: Table 4 (a) shows the contribution of each component of our method to the overall solution. The experiment was conducted on the LLFF dataset using the exact same settings reported for Table 1.

As can be seen, changing even a single component of the full method causes dramatic performance degradation. Not surprisingly, the most significant degradation happened when we disabled the initialization technique to find T_0 . NeRF tends to be accurate in the region of interest, but produces completely meaningless results outside it. Hence, without initialization the photometric loss in this case is useless. Another major performance degradation occurred when no viewshed maps were used. This can be explained by looking at our viewshed-based sampling example, shown in Figure 5. When generating novel views of unknown NeRF scenes it is very likely to capture some areas out of the original region of interest, which is basically noise in terms of NeRF. The viewshed maps filter



Fig. 7: Illumination: (Left) 4 images from the original videos, top images taken from one video and the bottom images from another video. Each video was taken under different illumination condition. (Right) Registration of the two scenes where the bottom left part is taken from one NeRF and the top right part is taken from another NeRF.

out the noisy pixels and thus the photometric loss (Equation 7) is applied to more reliable pixels.

Noisy VF: In the next ablation study we add noise to the oriented points used to construct VF. Specifically, we add up to 20% (of the size of the scene) zero-mean noise to the position of the oriented points that are used to train the NF algorithm. Then, we generated camera viewpoints by sampling the NF latent space, as before. Table 4 (b) shows the results. It is interesting to note that even when noise of up to 20% of the size of scene is added to the position of the oriented points, performance do not degrade. We suspect this is because the goal of the oriented points, that are fed to the Normalizing Flows algorithm, is simply to generate a plausible *initialization* for the position of the virtual cameras. This is enough to produce sufficiently good novel images for the optimization.

Illumination changes Our method minimizes a photometric loss, it is therefore a fair question to ask if it works with different illumination conditions? To test that we captured a scene twice, each time with different illumination, and then reconstructed a NeRF for each instance. We ran COLMAP on each video separately to extract the pose estimation, and found that this has a several consequences. First, each NeRF has its own coordinate system so there is no ground-truth and the results are qualitative. Second, each NeRF has its own arbitrary scale factor since COLMAP only estimates extrinsics and intrinsics up to scale. In order to deal with this issue, we added a learnable scale factor to our method on top of the 6-DoF learnable parameters. This is a generalization of the base method for the case of scenes with unknown scale. Our qualitative results are shown in Figure 7. On the left we show two frames from each of the videos. Observe how the color of the table changes, as a result of illumination change. On the right we show the scene from a novel view point where images from both NeRFs are registered using VF-NeRF. As can be seen, the registration is quite accurate, despite the change in illumination. This is especially important because VF-NeRF minimizes a photometric loss.



Fig. 8: Failure Example: Failure example of a scene from Objaverse dataset that converged to a wrong solution. The registered scene is upside-down with respect to the original scene. The image is from a novel view point and is created from the two registered NeRFs where we alternate between them using a checkerboard pattern of 50 pixels (zoom in to see the pattern artifacts on the skateboard).

Limitations VF-NeRF registration relies on photometric loss, a method susceptible to inaccuracies when applied to textureless surfaces. Moreover, VF-NeRF may converge to a partially accurate solution in terms of photometric loss, however, in scenes exhibiting high symmetry, this solution may deviate by 180° from the true solution, as illustrated in Figure 8. It would also be interesting to stress test VF-NeRF on scenes with partial overlap, or scenes with moving objects.

Why not use COLMAP? It is certainly possible to use COLMAP from scratch to register images from both NeRFs. But, we believe that NeRF registration can provide an efficient building block for modular and scaleable pipeline, where small groups of images are bundle adjusted together into small NeRFs that are later registered into larger NeRF models. In addition, Viewshed Fields, are quite a useful tool for sampling "good" novel view points as well as sampling "good" 3D points of the scene. Going forward, there might be novel applications that will benefit from NeRF registration in general, and VF in particular.

5 Conclusion

We considered the problem of NeRF registration and suggested a novel representation, termed Viewshed Fields (VF), to help solve it. VF is an implicit function, just like NeRF, that captures the likelihood of 3D surface points to be viewed by the original cameras. The novel combination of VF with Normalizing Flows (NF) helps in various use cases. It can be used to sample novel camera view points, on one hand, or to sample a colored 3D point cloud, on the other. It can also be used to guide ray sampling during the optimization of NeRF registration. We evaluated our method, VF-NeRF, on several diverse datasets that include front-facing scenarios, object-centric videos, and images of synthetic objects, and achieved SOTA results on many of them.

6 Supplementary

7 Appendix Overview

This document contains supplemental material regarding the following topics:

1. **Appendix Overview**
2. **Original Images** - Discussion about the differences between using generated novel views and the original images.
3. **Noise Robustness Study** - Visualization of the noise robustness study.
4. **VF Direction** - Discussion and visualization of the importance of direction d to the VF
5. **Registration Examples** - Videos of VF-NeRF registration results of different scenes.
6. **"Casually Captured" Dataset** - Examples from our casually captures scenes.
7. **Implementation Details** - Full detailed implementation details including computation, parameters etc.
8. **Full Results** - The results of all the experiments, represented scene by scene.

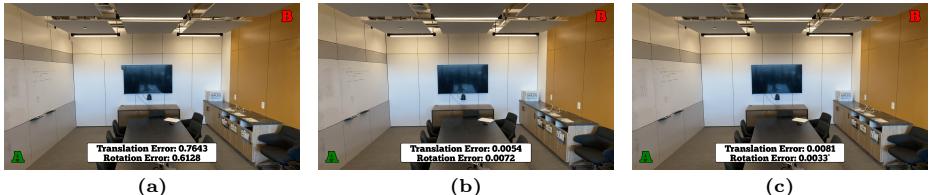


Fig. 9: Error Visualization: A visualization of the registration results of different methods, this visualization clarifies the visual impact behind the RMS error numbers. 9a is INeRF based (Observe the mismatch on the edge of the TV), 9b is our method, 9c is original cameras based, all three initialized with our VF. Note that the difference between 9b and 9c is visually indistinguishable.

8 Original Images

Our method is completely agnostic to the original images that were used to generate the NeRF 3D scene, but it is a fair question to ask - why not use them for the registration process instead of using NF and viewshed maps?

To address this question we evaluated an extensive experiment over the LLFF dataset 5 comparing our method to registration based on the original images, the experiment settings are exactly the same as the LLFF dataset experiment. Table 5 shows that we can get comparable results with both the original images

and the generated VF images. Moreover, we argue that although the error is numerically lower when using the original images, it is visually indistinguishable as shown in figure 9.

An explanation for the numerical difference is that we use COLMAP as a ground truth for determining camera poses, which is subject to estimation inaccuracies too. the inaccuracies might be different between the methods since COLMAP minimizes **geometric** objective function, while our method minimizes **photometric** objective function. Thus, in scenarios with minimal registration error within a noisy setting, discerning the superiority between COLMAP’s error and the photometric error remains ambiguous.

Table 5: Using Viewshed Fields Vs. Original Images: We compare registration error when using the original images vs. using VF-NeRF. Comparison is done on the LLFF dataset in three different settings and the results are the mean error over all the scenes in the dataset. Δt denotes the RMS translation errors multiplied by 1e2. As can be seen, the error of both methods is under 0.05 for rotation and translation in all categories.

Model	$\Delta t \downarrow$	$\Delta R \downarrow$	$\Delta t \downarrow$	$\Delta R \downarrow$	$\Delta t \downarrow$	$\Delta R \downarrow$
	Full Overlap	Partial Overlap	No Overlap			
Original images + Photo init	0.0132	0.0099	0.0182	0.0089	0.0188	0.0143
VF-NeRF + Photo Init.	0.0151	0.0206	0.0393	0.0358	0.0324	0.0358

9 Noise Robustness Study

Figure 10 illustrates the impact of introducing noise to oriented points through point cloud visualization. A noticeable decline in quality is observed when comparing the point clouds of the normal and noised scenes. However, it also reveals that the generated oriented points from the noised scene are capable of generating a reasonably accurate point cloud, thus enabling the creation of satisfactory novel views, as demonstrated in the noise robustness study section of the main paper.

10 VF direction

Viewshed Fields (VF) maps oriented point (\mathbf{x}, \mathbf{d}) to latent space \mathbf{z} . When looking at a generated VF map, it is clear that oriented points with high values are located (*i.e.* \mathbf{x}) in the ROI, and oriented points with low values are located outside the ROI. But, looking at a single image is not enough to understand the importance of the direction \mathbf{d} . To clarify the important effect of the direction \mathbf{d} to the VF, we generated VF of the same object from different directions.



Fig. 10: Noise Robustness Study Point Clouds: Two point clouds of the Trex (LLFF) scene generated by VF samples. (Left) A Point cloud from a scene without noised oriented points. (Right) A Point cloud from a scene with noised oriented points (20% of the scene scale with zero-mean distribution).

Figure 11 demonstrates the crucial effect of direction on the VF, which supports the novel view generation by estimation of a good-looking direction. For a better understanding of the effect of direction on the VF, see the attached video.

11 Registration Examples

Throughout the paper, we visualize the registration results using novel views and point clouds. We use videos to show the complete scene registration, as they cover the whole scene from different viewpoints. See the attached videos for registration examples on three different scenes. Note that there are some artifacts ("floaters") in the videos, these artifacts are not related to the registration process but rather to the NeRF quality of some novel views, as expected when one of the scenes is not originally well covered from the novel viewpoint.

12 "Casually Captured" Dataset

We used our own captured scene called "casually captured" to demonstrate the registration task on naturally captured scenes. Since this data is new, we show some examples from each scene. Figure 12 shows frames from the "Lion" scene from the "Casually captured" dataset and Figure 13 shows frames from the "Table" scene from the "Casually captured" dataset.

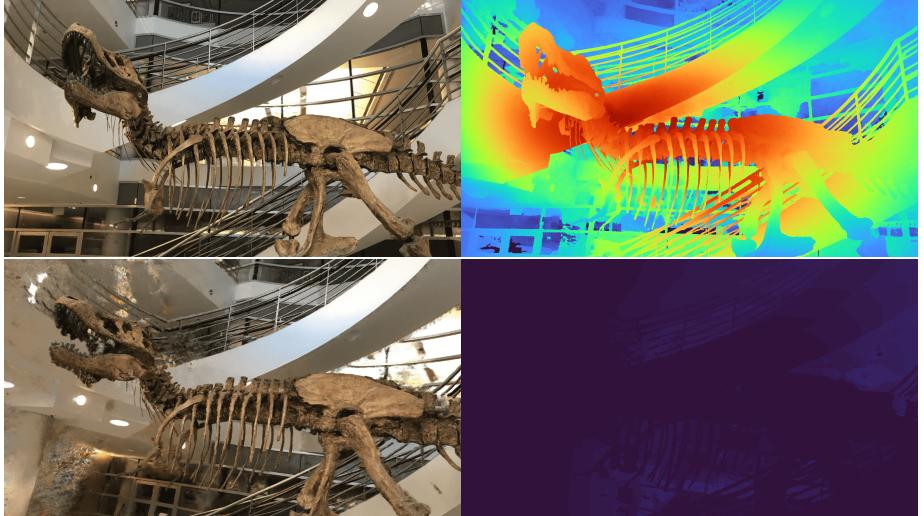


Fig. 11: Viewshed Fields Direction: (Top) Example of novel view and its corresponding VF map from the cameras plane of LLFF Trex scene, which is a good looking direction (Bottom) Example of novel view and its corresponding VF map from the side of LLFF Trex scene, which is a bad looking direction. Note that the oriented points are in the same location \mathbf{x} but different direction \mathbf{d} with a major affect on the image quality and VF values as expected.

13 Implementation Details

We used NVIDIA A5000 GPU for all the experiments, working on a single scene each time. As our NeRF representation we used Nerfacto [41], to train NeRF we sample 1024 rays each iteration, using Adam [22] as an optimizer with an initial learning rate of $1e^{-2}$ and exponential decay.

To learn the VF we follow Real-NVP [13] with $L=4$ layers and $H=128$ hidden dims, as an optimizer we use RAdam [27] with a constant learning rate of $5e^{-5}$. The real scene NeRFs trained for 60K iterations and the VF train is enabled on the last 10K iterations. The synthetic scene NeRFs are trained for 20K iterations and the VF train is enabled on the last 5K iterations and ignored where the image is transparent ($\alpha < 128$ for RGBA images).

Photometric initialization is done over 25 random transformations. For PC initialization we first generate point clouds by sampling 100K points from the VF distribution, choose the points with a density higher than 10, and use these point clouds as input for the classic global registration method. In our case, we use FPFH [35] for feature extraction.

As for the registration phase, for the real scenes, we optimize the 6DoF parameters for 15K iterations with 32K samples per iteration, we use SGD optimizer with an initial learning rate of $5e^{-3}$ and exponential decay. For the synthetic scenes, we optimize the 6DoF parameters for 2.5K iterations with 8128



Fig. 12: Lion Scene: Four frames from the casually captured "Lion" scene



Fig. 13: Table Scene: Four frames from the casually captured "Table" scene

samples per iteration, we use SGD optimizer with an initial learning rate of $1e-3$ and exponential decay.

14 Full Results

Table 6 shows the full results over LLFF dataset and Table 7 shows the full results over Objaverse dataset. That is the raw transformation and rotation error for each scene and each model.

Table 6: LLFF Results: Rotation and translation RMS error comparison on LLFF dataset, divided into the three experiment types regarding overlap between the frames. Δt denotes the RMS translation errors multiplied by $1e2$. FPFH + RANSAC uses point clouds generated by Viewshed Fields. VF-NeRF + PC Init refers to our algorithm after initializing with VF-based points clouds using FPFH + RANSAC. VF-NeRF + Photo Init refers to our method using photometric based initialization.

Model	Fern		Horns		Room		Trex	
	$\Delta t \downarrow$	$\Delta R \downarrow$						
Full Overlap								
FPFH [35] + RANSAC [15]	0.9681	1.3997	1.7415	1.2832	2.5040	1.9991	3.2461	1.4443
iNeRF [46]	8.8749	10.4430	19.2539	19.6978	38.8494	3.2129	24.5847	31.1075
iNeRF [46] + Photo Init.	0.0354	0.0214	0.0385	0.0055	0.7643	0.6128	0.0634	0.0098
VF-NeRF + Photo Init.	0.0371	0.0430	0.0107	0.0201	0.0054	0.0072	0.0073	0.0122
VF-NeRF + PC Init.	0.0321	0.0276	0.0078	0.0152	0.0045	0.0076	0.0205	0.0122
Partial Overlap								
FPFH [35] + RANSAC [15]	9.6166	5.6116	1.6095	1.6392	2.0541	1.8564	1.3781	1.7431
iNeRF [46]	8.3334	14.5468	11.6589	21.2608	0.0478	0.0199	61.2590	14.2920
iNeRF [46] + Photo Init.	0.0583	0.0235	51.9557	9.9470	0.0280	0.0248	0.0659	0.0252
VF-NeRF + Photo Init.	0.0885	0.1021	0.0301	0.0219	0.0153	0.0145	0.0232	0.0045
VF-NeRF + PC Init.	0.0724	0.0891	0.0179	0.0127	0.0335	0.0233	0.0189	0.0130
No Overlap								
FPFH [35] + RANSAC [15]	3.6367	2.9586	1.0247	1.6241	3.2338	3.1353	1.1763	3.5331
iNeRF [46]	16.0154	13.9615	0.0242	0.0164	10.6372	12.0546	8.9484	14.3406
iNeRF [46] + Photo Init.	0.0678	0.0364	11.1795	22.5305	0.0435	0.0166	0.0493	0.0219
VF-NeRF + Photo Init.	0.0787	0.1029	0.0325	0.0183	0.0150	0.0063	0.0035	0.0157
VF-NeRF + PC Init.	0.0544	0.0545	0.0225	0.0246	0.0123	0.0247	0.0102	0.0105

Table 7: Quantitative results of registration on the Objaverse dataset. $\Delta\mathbf{R}$ denotes the relative rotation errors in degree, $\Delta\mathbf{t}$ denotes the relative translation errors multiplied by $1e2$ with unknown scales. DReg_{df} refers to DReg with density fields and DReg refers to DReg with surface field. FPFH + RANSAC uses point clouds generated by Viewshed Fields. VF-NeRF + PC Init refers to our algorithm after initializing with VF-based point clouds using FPFH + RANSAC. The results of FGR, REGTR and DReg are taken from [6].

	Food 5648 Chair 4b05 Chair 4659 Chair 3f2d Cone 37b5 Figurine 260d Figurine 0a5b Figurine 09f0 Banana 3a07 Banana 2373 Banana 0a07											
FGR [51]	178.34	50.50	28.54	81.31	104.52	89.13	26.35	138.00	12.17	6.92	2.86	
$\Delta\mathbf{R}$	REGTR [47]	169.07	150.38	92.80	98.67	62.50	111.80	106.12	176.48	136.02	178.36	173.96
DReg _{df} [6]	77.48	160.13	157.21	22.91	108.09	121.32	10.53	95.89	95.43	3.49	6.96	
DReg [6]	6.01	6.53	17.74	18.88	18.79	2.11	7.62	8.25	15.55	10.95	1.36	
FGR [51] + RANSAC [15]	3.49	3.11	1.91	3.04	5.47	4.31	2.77	9.39	2.06	1.49	1.74	
VF-NeRF + PC Init.	0.03	0.03	0.03	0.80	49.05	0.84	0.60	0.92	0.03	1.28	0.03	
FGR [51]	17.44	2.27	7.10	8.65	30.49	19.25	10.93	35.22	8.50	1.53	1.36	
$\Delta\mathbf{t}$	REGTR [47]	30.72	15.41	24.97	60.53	84.20	62.07	35.48	42.10	10.75	50.40	13.17
DReg _{df} [6]	15.52	7.32	11.72	2.29	21.70	33.61	1.95	21.40	13.14	4.28	0.50	
DReg [6]	1.78	4.13	8.74	5.07	3.06	3.54	10.68	3.18	0.46	1.00	1.22	
FPFH [33] + RANSAC [15]	0.67	0.42	3.41	0.45	23.96	2.82	0.55	3.16	0.50	0.13	3.01	
VF-NeRF + PC Init.	0.16	2.43	0.05	1.33	23.73	0.26	0.005	0.16	0.27	0.15	0.44	
	Firepug 0e6ds Firepug 0e06s Firepug 0e063 Firepug 0152 Shoe 1e3 Shoe 1e27 Shoe 0b9 Show 022e-Teddy 1e47 Elephant 1e3a Shoe 1e3a											
FGR [51]	6.19	20.32	7.50	10.23	178.14	71.55	50.28	8.05	7.65	21.37	30.97	
$\Delta\mathbf{R}$	REGTR [47]	156.92	99.60	4.04	2.55	175.21	97.92	154.91	149.17	177.15	172.28	102.62
DReg _{df} [6]	156.13	45.76	3.32	14.69	156.56	156.34	6.32	6.97	3.70	126.94		
DReg [6]	7.96	17.43	4.86	6.06	12.95	6.48	2.03	11.14	8.00	11.13	13.84	
FPFH [33] + RANSAC [15]	5.65	2.06	1.53	2.07	14.38	6.44	2.96	7.14	1.23	8.30	1.28	
VF-NeRF + PC Init.	0.36	0.03	1.11	1.06	5.38	0.03	6.07	1.09	3.25	0.71		
FGR [51]	5.83	0.83	1.17	0.04	4.99	8.82	35.47	1.11	4.51	14.08	11.03	
$\Delta\mathbf{t}$	REGTR [47]	68.71	38.74	2.13	3.53	43.40	61.37	102.00	42.84	52.26	66.15	34.54
DReg _{df} [6]	10.54	5.32	2.60	4.66	28.63	24.82	4.40	2.20	4.26	1.40	33.57	
DReg [6]	1.58	5.08	0.96	2.08	12.80	1.81	0.65	1.06	8.97	6.17	7.80	
FPFH [33] + RANSAC [15]	0.47	0.17	2.98	1.33	7.40	1.08	2.05	6.54	0.30	3.88	0.86	
VF-NeRF + PC Init.	0.006	0.023	2.60	0.24	0.13	4.56	0.19	5.90	0.17	2.57	0.76	
	Piano 0e0d Piano 0a6c Truck 1e43 Guitar 15b4 Guitar 14b8 Guitar 0cc6 Guitar 0aa0 Lantern 0231 Lamp 0230 Bench 0b05 Shield 22a7											
FGR [51]	23.09	77.63	7.46	7.80	5.25	13.07	39.94	130.36	17.44	19.51	170.27	
$\Delta\mathbf{R}$	REGTR [47]	30.54	117.90	178.49	5.18	29.47	103.84	5.05	139.32	100.45	122.12	157.38
DReg _{df} [6]	160.77	165.79	117.07	11.43	16.67	177.62	7.56	7.76	173.09	179.16	170.90	
DReg [6]	16.30	13.51	16.68	12.60	3.43	1.08	9.53	9.17	16.44	12.98	8.21	
FPFH [33] + RANSAC [15]	5.80	12.86	6.62	1.84	3.91	2.06	2.44	2.69	2.82	2.47	1.29	
VF-NeRF + PC Init.	6.89	6.37	0.03	1.34	1.98	1.16	1.54	0.83	0.03	0.03	3.68	
FGR [51]	7.43	14.50	5.95	2.86	3.46	1.83	8.42	9.06	0.69	12.52	15.57	
$\Delta\mathbf{t}$	REGTR [47]	44.24	65.99	50.63	15.18	18.41	89.20	9.91	57.25	64.44	31.97	44.29
DReg _{df} [6]	22.86	26.18	24.83	10.27	8.50	43.21	4.09	3.35	26.77	28.59	34.71	
DReg [6]	4.80	12.54	0.04	5.72	5.03	3.01	1.20	3.29	1.31	1.68	11.33	
FPFH [33] + RANSAC [15]	3.74	3.14	4.27	2.15	1.05	1.47	1.16	2.43	0.91	5.43	3.07	
VF-NeRF + PC Init.	13.84	6.45	0.36	0.14	0.14	0.74	0.16	0.36	0.11	0.27	8.03	
	Shield 1e7a Shield 1e8b Shield 0b0d Controller 0e06 Fighter Jet 1e6e Fighter Jet 0e0f Fighter Jet 0e09 Telephone 1e3c Telephone 0e34 Lampshade 0a6b Skateboard 1e0-7											
FGR [51]	198.83	175.99	7.06	10.01	11.21	46.21	59.56	150.10	18.76	147.50	176.90	
$\Delta\mathbf{R}$	REGTR [47]	138.94	169.78	14.76	102.95	154.74	158.64	178.35	144.47	1.13	148.44	3.88
DReg _{df} [6]	178.93	4.92	7.78	179.13	9.75	23.97	178.68	132.49	16.59	5.70	179.97	
DReg [6]	12.94	12.26	2.29	4.03	6.88	10.53	6.46	15.60	9.01	5.67	1.92	
FPFH [33] + RANSAC [15]	9.62	13.62	1.14	7.14	1.50	3.55	1.31	23.12	3.32	1.98	179.45	
VF-NeRF + PC Init.	2.74	19.17	1.07	0.26	0.09	1.12	0.28	0.03	1.19	1.10	179.10	
FGR [51]	49.41	55.62	0.22	5.97	6.90	11.73	3.44	57.53	13.51	67.08	19.03	
$\Delta\mathbf{t}$	REGTR [47]	72.01	48.56	6.57	65.87	52.97	82.10	28.79	51.09	0.91	54.45	5.14
DReg _{df} [6]	50.66	0.66	1.58	23.68	2.64	13.98	19.28	63.76	15.65	4.81	12.20	
DReg [6]	2.79	3.9	1.26	0.99	2.53	7.55	1.59	1.28	3.57	0.84	0.46	
FPFH [33] + RANSAC [15]	1.52	4.40	2.65	4.12	1.18	3.97	4.10	8.48	1.03	3.07	3.04	
VF-NeRF + PC Init.	1.61	6.30	2.65	1.96	0.15	0.19	0.26	0.65	1.45	1.45	1.93	

References

1. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-9**(5), 698–700 (1987). <https://doi.org/10.1109/TPAMI.1987.4767965> 3
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5855–5864 (October 2021) 3
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022) 3
4. Besl, P., McKay, N.D.: A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(2), 239–256 (1992). <https://doi.org/10.1109/34.121791> 3
5. Bongsoo Choy, C., Stark, M., Corbett-Davies, S., Savarese, S.: Enriching object detection with 2d-3d registration and continuous viewpoint estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015) 1
6. Chen, Y., Lee, G.H.: Dreg-nerf: Deep registration for neural radiance fields (2023) 4, 11, 21
7. Chen, Y., Chen, X., Wang, X., Zhang, Q., Guo, Y., Shan, Y., Wang, F.: Local-to-global registration for bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8264–8273 (2023) 4
8. Chetverikov, D., Svirko, D., Stepanov, D., Krsek, P.: The trimmed iterative closest point algorithm. In: 2002 International Conference on Pattern Recognition. vol. 3, pp. 545–548. IEEE (2002) 3, 4
9. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2514–2523 (2020) 4
10. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects (2022) 8, 11
11. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12882–12891 (2022) 3
12. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516 (2014) 2, 6
13. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real NVP. In: International Conference on Learning Representations (2017), <https://openreview.net/forum?id=HkpbnH9lx> 5, 6, 18
14. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3d object recognition. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 998–1005. Ieee (2010) 3
15. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (jun 1981). <https://doi.org/10.1145/358669.358692>, <https://doi.org/10.1145/358669.358692> 9, 10, 11, 20, 21

16. Fu, X., Zhang, S., Chen, T., Lu, Y., Zhu, L., Zhou, X., Geiger, A., Liao, Y.: Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In: 2022 International Conference on 3D Vision (3DV). pp. 1–11. IEEE (2022) [3](#)
17. Garbin, S.J., Kowalski, M., Johnson, M., Shotton, J., Valentin, J.: Fastnerf: High-fidelity neural rendering at 200fps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14346–14355 (2021) [3](#)
18. Goli, L., Rebain, D., Sabour, S., Garg, A., Tagliasacchi, A.: nerf2nerf: Pairwise registration of neural radiance fields. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 9354–9361 (2023). <https://doi.org/10.1109/ICRA48891.2023.10160794> [4](#)
19. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., Kwok, N.M.: A comprehensive performance evaluation of 3d local feature descriptors. International Journal of Computer Vision **116**, 66–89 (2016) [3](#)
20. Hezroni, I., Drory, A., Giryes, R., Avidan, S.: Deepbbs: Deep best buddies for point cloud registration. In: 2021 International Conference on 3D Vision (3DV). pp. 342–351. IEEE (2021) [4](#)
21. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/> [3](#)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017) [18](#)
23. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31** (2018) [5](#)
24. Li, P., Wang, R., Wang, Y., Tao, W.: Evaluation of the icp algorithm in 3d point cloud registration. IEEE Access **8**, 68030–68048 (2020) [3](#)
25. Li, Z., Wang, Q., Cole, F., Tucker, R., Snavely, N.: Dynibar: Neural dynamic image-based rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4273–4284 (June 2023) [3](#)
26. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5741–5751 (October 2021) [4](#)
27. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J.: On the variance of the adaptive learning rate and beyond (2021) [18](#)
28. Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7210–7219 (June 2021) [5](#)
29. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) (2019) [8](#), 9
30. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (dec 2021). <https://doi.org/10.1145/3503250>, <https://doi.org/10.1145/3503250> [2,3](#)
31. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). <https://doi.org/10.1145/3528223.3530127>, <https://doi.org/10.1145/3528223.3530127> [3](#)
32. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) [4](#)

33. Pomerleau, F., Colas, F., Siegwart, R.: A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics* **4**(1), 1–104 (2015). <https://doi.org/10.1561/2300000035>, <http://dx.doi.org/10.1561/2300000035> 1
34. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: Proceedings third international conference on 3-D digital imaging and modeling. pp. 145–152. IEEE (2001) 3
35. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: 2009 IEEE International Conference on Robotics and Automation. pp. 3212–3217 (2009). <https://doi.org/10.1109/ROBOT.2009.5152473> 3, 9, 10, 11, 18, 20, 21
36. Schönberger, J.L., Frahm, J.M.: Structure-from-Motion Revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 4, 9
37. Shen, J., Agudo, A., Moreno-Noguer, F., Ruiz, A.: Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification. In: European Conference on Computer Vision. pp. 540–557. Springer (2022) 5
38. Shen, J., Ruiz, A., Agudo, A., Moreno-Noguer, F.: Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. In: 2021 International Conference on 3D Vision (3DV). pp. 972–981. IEEE (2021) 5
39. Sinko, M., Kamencay, P., Hudec, R., Benco, M.: 3d registration of the point cloud data using icp algorithm in medical image analysis. In: 2018 ELEKTRO. pp. 1–6 (2018). <https://doi.org/10.1109/ELEKTRO.2018.8398245> 1
40. Sünderhauf, N., Abou-Chakra, J., Miller, D.: Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 9370–9376. IEEE (2023) 5
41. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23 (2023) 18
42. Tombari, F., Salti, S., Di Stefano, L.: Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision* **102**(1-3), 198–220 (2013) 3
43. Tschernezki, V., Laina, I., Larlus, D., Vedaldi, A.: Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In: 2022 International Conference on 3D Vision (3DV). pp. 443–453. IEEE (2022) 3
44. Vora, S., Radwan, N., Greff, K., Meyer, H., Genova, K., Sajjadi, M.S., Pot, E., Tagliasacchi, A., Duckworth, D.: Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. arXiv preprint arXiv:2111.13260 (2021) 3
45. Wang, Y., Solomon, J.M.: Deep closest point: Learning representations for point cloud registration. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3523–3532 (2019) 4
46. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: iNeRF: Inverting neural radiance fields for pose estimation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2021) 4, 8, 9, 10, 20
47. Yew, Z.J., Lee, G.H.: Regtr: End-to-end point cloud correspondences with transformers (2022) 11, 21
48. Zhang, X., Yang, J., Zhang, S., Zhang, Y.: 3d registration with maximal cliques (2023) 4
49. Zheng, Y., Li, Y., Yang, S., Lu, H.: Global-pbnet: A novel point cloud registration for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems* **23**(11), 22312–22319 (2022). <https://doi.org/10.1109/TITS.2022.3153133> 1

50. Zhi, S., Laidlow, T., Leutenegger, S., Davison, A.J.: In-place scene labelling and understanding with implicit scene representation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15838–15847 (2021) [3](#)
51. Zhou, Q.Y., Park, J., Koltun, V.: Fast global registration. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14. pp. 766–782. Springer (2016) [4](#), [11](#), [21](#)
52. Zinßer, T., Schmidt, J., Niemann, H.: A refined icp algorithm for robust 3-d correspondence estimation. In: Proceedings 2003 international conference on image processing (Cat. No. 03CH37429). vol. 2, pp. II–695. IEEE (2003) [3](#)