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

Reg-NF: Efficient Registration of Implicit Surfaces within Neural Fields

Stephen Hausler[1,†], David Hall[1,†], Sutharsan Mahendren[1,2], Peyman Moghadam[1,2]

***Abstract*— Neural fields, coordinate-based neural networks, have recently gained popularity for implicitly representing a scene. In contrast to classical methods that are based on explicit representations such as point clouds, neural fields provide a continuous scene representation able to represent 3D geometry and appearance in a way which is compact and ideal for robotics applications. However, limited prior methods have investigated registering multiple neural fields by directly utilising these continuous implicit representations. In this paper, we present Reg-NF, a neural fields-based registration that optimises for the relative 6-DoF transformation between two arbitrary neural fields, even if those two fields have different scale factors. Key components of Reg-NF include a bidirectional registration loss, multi-view surface sampling, and utilisation of volumetric signed distance functions (SDFs). We showcase our approach on a new neural field dataset for evaluating registration problems. We provide an exhaustive set of experiments and ablation studies to identify the performance of our approach, while also discussing limitations to provide future direction to the research community on open challenges in utilizing neural fields in unconstrained environments.**

I. Introduction

For robotics applications, the six degree of freedom (6-DoF) registration between two scenes of interest is a crucial step, for tasks such as localisation, object pose estimation and 3D reconstruction. While many methods exist for representing 3D scenes, including point clouds, voxels and meshes, recently *implicit representations* have emerged, which can compactly represent 3D scenes with unprecedented fidelity. Recent implicit representations are typically expressed as neural fields (NFs) and are created using differentiable volumetric rendering [1]. This was first popularised by the neural radiance field (NeRF) [1] and while it and follow-up works [2]–[4] have shown impressive visual rendering performance, their underlying surface geometry has a limited fidelity [5] and are not smooth [6]. Recently, NeuS [5] has shown how volumetric rendering can be used to train signed distance functions (SDFs) [7], enabling neural fields with highly accurate geometric representations better suited for registration.

Neural field registration is important for their use in robotic applications, as it enables uses such as the fusion of multiple implicit maps, and the ability to dynamically update an existing implicit field. Nerf2nerf [8] was one of the first works to investigate neural field registration, by considering the registration problem as a deep learnt optimisation function between two neural fields. However, nerf2nerf relies on human-annotated keypoints for initialisation and assumes the scale of two neural fields are the same. In this paper, we propose Reg-NF, a novel method to estimate the relative 6-DoF pose transformation between two objects of interest which are located in two different neural fields. Our proposed method does not rely on human-annotated keypoints, operates directly on the continuous neural fields, and is capable of estimating transformation between two models with arbitrary scales. We build on nerf2nerf, proposing a bidirectional registration loss, the use of multi-view sampling of the NF surface, and the use of SDFs as the implicit model of choice. These increase registration accuracy, take advantage of NFs ability to render data from any view, and ensure a consistent and clear geometric representation of implicit models respectively.

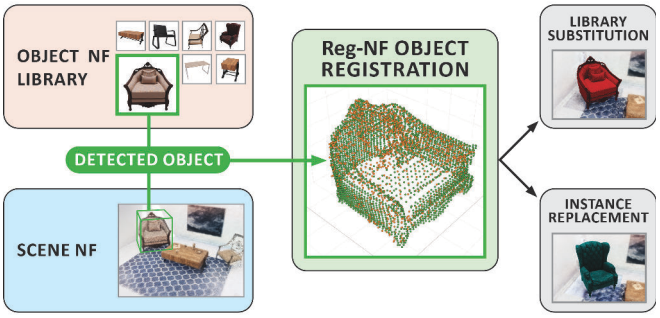


Fig. 1. The proposed pipeline for using Reg-NF registration. Object is detected in a scene neural field (NF) and matched to an object in a library of object-centric NF. Reg-NF performs registration between the two NFs enabling neural substitution of the library object NF into the scene or the replacement of the object with another from the library. Substitution and replacement models coloured for clarity.

Our experiments highlight use-cases for NF registration where the 6DoF output from Reg-NF can be used for NF editing to merge NFs together. We consider the scenario where Reg-NF is used to register a large scene NF with high-fidelity object-centric NFs stored in an object library, enabling both substitution of library objects into the scene, and replacement of object instances within the scene as shown in Fig. 1. This shows two particular benefits for Reg-NF in robotics. The first is object completion, using library substitution to improve the representation of scenes that have only partially observed objects or have under-trained NFs due to hardware constraints. The second is using instance replacement as a way to enable data-driven simulation where any scene NF can be edited (by replacing objects instances) and used as new data for training in simulated NF environments.

† Equal Contribution. Website: https://csiro-robotics.github.io/Reg-NF

[1] Authors are with the CSIRO Robotics, DATA61, CSIRO, Brisbane, QLD 4069, Australia. E-mails: firstname.lastname@csiro.au

[2] Sutharsan Mahendren, and Peyman Moghadam are with the SAIVT research programme in the School of Electrical Engineering and Robotics, Queensland University of Technology (QUT), Brisbane, Australia. E-mails: {sutharsan.mahendren, peyman.moghadam}@qut.edu.au

II. Related Work

A. Neural Fields

Neural fields (NFs) are implicit representations of 3D space that have gained great popularity in recent years [1]–[5], [9]–[11]. NFs use a small neural network to map any point in a normalised 3D space $\mathbf{x} \in \mathbb{R}^3$ to the field's desired output/s (*e.g.*, colour, density, opacity, etc.).

The recent popularity of neural fields is attributed to the introduction of Neural Radiance Fields (NeRFs) [1] which has spawned many radiance field derivatives [3], [11], [12]. The NeRF model is a neural field which maps $\mathbf{x}$ and a viewing angle $\mathbf{v} \in \mathbb{R}^2$ to view-dependant colour $\mathbf{c}(\mathbf{x}, \mathbf{v}) \in \mathbb{R}^3$ and view-independent density $\sigma(\mathbf{x}) \in \mathbb{R}$. Density represents the likelihood of $\mathbf{x}$ hitting anything in space. With this model, any pixel can be represented as a ray ($\mathbf{r}$) described by origin ($\mathbf{o}$) and normalised orientation ($\mathbf{v}$), and the colour of that pixel can be computed through volumetric rendering [1]. NeRF can be trained using only images and associated camera poses by comparing the true colour of a given pixel to the one rendered from the NeRF.

Despite it's strengths, NeRF's focus is on visual rendering which can lead to inaccurate geometric representations [5], [9]. Surface neural network models (S) [5], [7], [9], [10], [13], seek to solve this by describing a continuous function f mapping $\mathbf{x}$ to the distance to the nearest surface $f(\mathbf{x}) \in \mathbb{R}$ with the surface located wherever $f(\mathbf{x}) = 0$. These are known to provide smooth and consistent geometric representations [5]. To enable training through purely image data as done by NeRF, some works combine the volumetric representation with implicit surface models [5], [9], [10] and it is these models that will be the focus within our work, outlined in more detail within Section III.

B. Neural Fields for Robotics

Unlike explicit scene representations, such as voxels, point clouds, meshes, and surfels [14], NFs have attractive properties for robotics as their representations are continuous and memory efficient. Recent work has extended the uses of NFs beyond synthesising novel views to robotics applications. Prior methods [15]–[17] demonstrate using the gradient of density from the NFs for collision-free robot motion planning. Several approaches [18]–[20] show how continuous representation of NFs can be combined with additional constraints to jointly optimise for grasp and motion planning.

Many algorithms have recently been developed to extend Neural Fields in simultaneous localisation and mapping (SLAM). iMAP [21] is the first method to show NeRF-enabled SLAM with the aid of depth measurements from RGB-D sensors to reconstruct room-size scenes in real-time. NICE-SLAM [22] introduces a hierarchical implicit representation to represent larger scenes, and NICER-SLAM [23] applies neural implicit representations for RGB-only SLAM and shows promising real-time properties.

C. Neural Fields Registration

Registration is the task of estimating the relative transformation between two 3D scene models. It has been extensively studied on explicit representations (*e.g.*, point clouds), whilst registering multiple implicit scene representations remains an underexamined challenge. Nerf2nerf [8] is the first work which demonstrates relative transformation estimation between two NeRF models. However, they rely on human-annotated keypoints for initialisation, which is not practical in robotics applications. DReg-NeRF [24] uses NeRF models then coverts them to an occupancy voxel grid to train 3D CNN followed by attention layers to learn the relations between the pairwise feature grids. DReg-NeRF estimates transformation on explicit and discrete representation. Zero-NeRF [25] performs image to image registration leveraging NeRF representations but in the image space as opposed to between 3D scenes. Moreover, all the exiting prior works assume the scale of the two models are the same. In reality, each model's coordinate frame will be normalised to fit their specific training conditions. This does not guarantee consistent scale, particularly when considering models trained to represent individual objects and those trained to represent scenes. By comparison, our proposed method does not rely on human-annotated keypoints, and is capable of estimating transformation between two models with arbitrary scales.

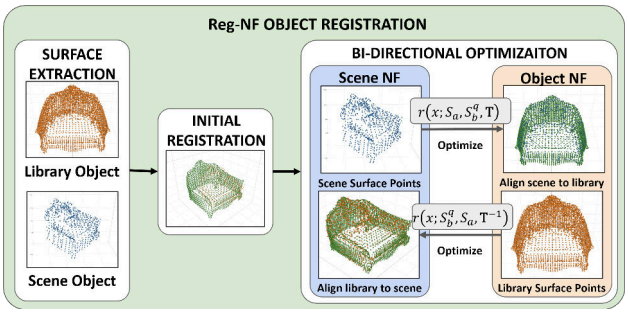


Fig. 2. Overview of our Reg-NF registration process. Blue and orange denotes surface sample points from the scene and library NFs for a matched object respectively. Green points represent the target alignment during optimisation. After surface extraction and an initial registration estimate, bi-directional optimisation iterates till convergence. Final output is a 6-DoF transformation matrix between models.

III. Methodology

A. Reg-NF Overview

In Reg-NF, we provide a technique for aligning the surfaces of two different SDF NFs, by minimising the difference between their surfaces values. Minimisation is optimised for the 6-DoF pose transformation $\mathbf{T}$ between the two NFs. We calculate $\mathbf{T}$ using a differentiable optimisation function, initialised with an automated procedure. While Reg-NF is generalisable across different use-cases, in this paper we aim to find $\mathbf{T}$ between a detected object in a larger scene NF with one in a pre-trained object NF library. We denote a as the notation of an implicit representation of a scene, and b^q as the qth object-specific implicit model from a database of object neural fields where $q \in \{1, ..., Q\}$. Please see Fig. 2 and Fig. 1 for an overview of Reg-NF and our considered use-case pipeline respectively.

B. Preliminaries

Our method assumes we have access to both scene and object NF models. As background information, we provide a

brief overview of the NF representation used for our work and its training process.

We utilise volumetric implicit surface fields, specifically NeuS [5] for all NFs in our work. These provide colour $c(\mathbf{x}, \mathbf{v})$ and signed surface distance $f(\mathbf{x})$ mapping functions derived from a shared backbone network with separate SDF and colour output heads.

For any given pixel in an image, points at distance (t) along the ray emitted from this pixel are represented as $\mathbf{r}(t) = \mathbf{o} + t\mathbf{v}|t \geq 0$. Given a sampling of points along a ray in the range $[t_1, t_2, ..., t_n]$ of increasing magnitude, volumetric rendering can be used to calculate the colour of a given pixel:

$$\hat{C}(\mathbf{r}, \mathbf{v}) = \sum_{i=1}^{n} T_i \phi_i c(\mathbf{r}(t_i), \mathbf{v}), \tag{1}$$

where T_i is the discrete accumulated transmittance at the ith point defined by $T_i = \prod_{j=1}^{i-1}(1 - \phi_j)$ and ϕ_i is the discrete opacity at the ith point along the ray. Discrete opacity in NeuS is akin to density in NeRF only it is calculated based on the neural SDF output at the given point along the ray.

Training of the model is done by randomly sampling a batch of (m) pixels from a set of training images with known camera poses such that we get training data $D = \{C(\mathbf{r}_k), \mathbf{o}_k, v_k\}$ from which n points are sampled along each ray. Loss is then computed as $L = L_c + \lambda L_e$ where L_c is the colour loss term defined by $L_c = \frac{1}{m}\sum_k |\hat{C}(\mathbf{r}_k) - C(\mathbf{r}_k)|$, L_e is an Eikonal loss regularisation term defined as $L_e = \frac{1}{nm}\sum_{k,i}(||f'(\mathbf{r}_k(t_i))||_2 - 1)^2$, and λ is a weighting factor for the Eikonal regularisation. For more details, refer to [5].

C. Automated initialisation

1) Initial multi-view surface sampling: Reg-NF begins by establishing approximate correspondences between objects of interest within a and b^q. We assume an object detection provides approximate location and classification of an object in a that matches to b^q. Based on this, we calculate a set of N camera locations (extrinsics, E_n), which are all oriented to view the centroid of the detected object of interest. We generate a grid pattern of rays travelling from each camera pose E_n and sample points along each ray. For each ray we return surface sample points at the first point along the ray where $f(\mathbf{x}) = 0$. These initial surface sample points for each of our N camera poses are then merged, providing object surface sample point sets P_a and P_b^q, for each object in our two respective neural fields. This multi-view sampling approach takes advantage of the ability of NFs to render data from any view, providing a clear geometric representation of the object for initialisation.

2) Calculate an initial transformation: To find the initial estimate of the transformation, we employ RANSAC [26] with Fast Point Feature Histogram (FPFH) [27] descriptors to estimate the correspondence between source and target. RANSAC provides an initial alignment approximation which is further refined via the point-to-point Iterative Closest Point (ICP) [28] method. Through this, for two initial sets of surface points P_a and P_b^q, we attain the initial six-degree-of-freedom (6-DoF) pose transformation $\hat{\mathbf{T}}$.

D. Bidirectional Registration Loss

We establish neural field registration as an optimisation function that uses gradient decent to find the optimal pose transformation $\mathbf{T}$ between two surface fields. The parameters of our optimisation are initialised using $\hat{\mathbf{T}}$, and assuming an initial scale factor of $s = 1$. Our neural field registration problem can be expressed by the following equation:

$$\arg\min_{\mathbf{T}} L_s\left(S_a, S_b^q; \mathbf{T}\right), \tag{2}$$

where S_a and S_b^q denotes the signed distance representation for implicit models a and b^q.

In nerf2nerf [8], optimisation is performed over a discrete set of samples collected from a single implicit model, with a robust kernel κ used to improve the robustness against outlier samples. The kernel contains learnt parameters p and α which control the decision boundary and impact of outlier samples on registration. In our approach, we consider that the accuracy of optimisation can be improved by collecting samples from both S_a and S_b^q (A and B^q respectively), and performing a bidirectional optimisation over these surfaces. Furthermore, we include a regulariser, which is designed to penalise the function when A and B^q are deviating from each other.

$$\begin{aligned} L_s(S_a, S_b^q; \mathbf{T}) = E_{x\in A}\kappa(r(x; S_a, S_b^q, \mathbf{T}); p, \alpha) \\ + E_{x\in B^q}\kappa(r(x; S_b^q, S_a, \mathbf{T}^{-1}); p, \alpha) + wL_r, \end{aligned} \tag{3}$$

where L_r is our regulariser with a weight factor w. Our loss function L_s is calculated using the current estimated pose transform $\mathbf{T}$, which we compose by multiplying three transformation matricies together, representing rotation, translation and scale factor components:

$$\mathbf{T} = T \cdot R \cdot \sigma, \tag{4}$$

where T denotes a 6-DoF translation transformation matrix with components $[t_x, t_y, t_z]$, R denotes a rotation transformation matrix composed from Euler angles $[r_r, r_p, r_y]$, and σ denotes a scale transformation matrix composed by scale factor s. Note that we use a scalar scale factor across all axes to avoid object warping. Our loss function is optimised over the learnt parameters: $(t_x, t_y, t_z, r_r, r_p, r_y, s, p, \alpha)$.

E. Registration Residuals and Sampling Procedure

Our registration residuals r are expressed as:

$$r(x; S_a, S_b^q, \mathbf{T}) = \|S_a - S_b^q\mathbf{T}\| \quad x \in A, \tag{5}$$

and in the bidirectional case:

$$r(x; S_b^q, S_a, \mathbf{T}^{-1}) = \|S_b^q - S_a\mathbf{T}^{-1}\| \quad x \in B^q. \tag{6}$$

Therefore, the residuals are calculating the difference in surface values between SDFs S_a and S_b^q. It is important to note that since SDF values are continuous, this loss formulation is differentiable. As SDF values increase in absolute magnitude the further they are away from a surface, the gradient descent will optimise the transformation $\mathbf{T}$ to match the surfaces even if the initial samples A and B^q have poor initialisation.

Algorithm 1: Reg-NF Sampling Procedure from S_a

$C \leftarrow A^{(t-1)} + \rho \cdot U_3[-1, +1]$;
for *each sample* $x \in C$ **do**
  if $S_a(x) \leq \omega_1$ ***and*** $S_b^q(x\boldsymbol{T}) \leq \omega_2$ ***and*** $r(x; S_a, S_b^q, \boldsymbol{T}) \leq \xi$ ***and*** $d(x, A^{(t-1)}) \geq \rho/10$ **then**
    $A_{(t)} \leftarrow A_{(t)} \cup \{x\}$;

Initially, we set samples A and B^q equal to the initial surface samples P_a and P_b^q. We allow for new samples to be discovered using a variant of the Metropolis-Hastings sampling procedure in nerf2nerf [8]. Our updated sampling algorithm is shown in Algorithm 1. Because SDF values are continuous and not discrete, the sampling strategy needs to be carefully designed to prevent the generation of samples away from true object surfaces.

F. Regularisation Loss

Our regularisation loss (L_r) term calculates the mean distance in 3D sample positions between the two different sets of sample points (one from each direction of optimisation); A, B^q. For point correspondence, we assume the nearest 3D point in the other set of sample points B^q is the ground-truth corresponding point given a particular sample point from A (similar to ICP [28]).

$$L_r = \sum \frac{\min_{B^q} D^2_{A,B^q}}{|A|}, D^2_{A,B^q} = \|A - B^q\mathbf{T}\|^2, \quad (7)$$

where $|A|$ denotes the number of samples in A. Given some sample points A from S_a, and a different set of transformed (by $\mathbf{T}$) sample points B^q from S_b^q, after optimisation, their 3D points should be approximately the same. The bidirectional components are balanced by L_r, which also acts to penalise 'unmatched points' between the two models. For example, if one set of samples covers the entirety of a large table in one SDF, and the second set of samples only considers a small proportion of the same table object in the second SDF, then a large disparity will be identified by the regulariser.

IV. Experimental Design

Dataset: The dataset we use for our experiment comprises high-fidelity simulated images and corresponding camera poses of objects and scenes, collected using NVIDIA's Omniverse Isaac Sim platform. We will refer to this dataset as our object NF registration (ONR) dataset. Object data is of single objects in a "void" and scene data is a standardised room with different object models placed within. Object data becomes the basis for our object NF library and contains data for 5 chair and 3 table models learnt at varying scales to maximize fidelity. Data for scenes is collected with manually generated trajectories under two conditions: long and short trajectories. Long trajectories maximise coverage of all objects within the scene while short ones do a more general pass of the room. These represent best-case and more realistic data capture for scene NF training respectively. There are three scenes collected: 1) containing all chair models stored as object data (ac-room); 2) containing all table models stored as object data (atbl-room); 3) a scene with a mixture of two chairs and a cluttered table (mix-room). Example images from each scene and individual object names are shown in Fig. 3.

(a) ac-room (b) atbl-room (c) mix-room

Fig. 3. Overview of scenes and object models in ONR dataset. Chairs objects shown in ac-room, starting clockwise from bottom-left are: chair (*c*), fancy chair with no pillow (*fc-nop*), matrix chair (*mc*), dining chair (*dc*), and fancy chair (*fc*). Tables shown in atbl-room from left to right are: willow table (wt), end table (et), and table (t). Objects of interest in mix-room are fc, t, and dc.

Metrics: We follow the metrics in nerf2nerf [8] for our work. We report the root mean squared error (RMSE) between the ground-truth and predicted scene to object transformation matrices. Rotation error $\Delta\mathbf{R}$ is calculated in radians and translation error $\Delta\mathbf{t}$ is calculated in normalised object frame units. We also report the absolute difference between estimated and true scale between scene and object models Δs. Note that this is given instead of RMSE as scale factor is assumed consistent across all axes.

Nerf2nerf on ONR dataset: To evaluate nerf2nerf on the ONR dataset, we enabled nerf2nerf to utilise the same surface fields from our SDF models as used by Reg-NF. The only modification to nerf2nerf was to modify their sampling algorithm to suit SDF notation, where a surface has a value of zero. We manually generated new human annotated keypoints for the initialisation procedure used in nerf2nerf.

Training models: All NF models were trained using the sdfstudio [30] implementation of NeuS which includes the proposal network from MipNeRF-360 [12] for training speed-up (neus-facto). For more details please refer to sdfstudio [30]. Object models were trained for 30,000 iterations and scene models were trained for 100,000 iterations unless stated otherwise for specific experiments.

Object Proposals: Reg-NF assumes a detection has already been made within a scene's NF through some pre-existing method. As object proposal generation is not within the scope of this work, we utilise ground-truth object 3D bounding boxes to calculate initial set of N camera extrinsics for generating the initial surface samples. We also remove any samples that are far away from the object's 3D location.

Reg-NF hyperparameters: We provide the following hyper-parameters for Reg-NF. For our sampler, we use $\omega_1 = 0.01$, $\omega_2 = 0.02$ and $\xi = 0.02$. We set ρ to $r/20$, where r is the scene radii and generate new samples every 10 iterations. We use a learning rate of 0.02 for rotation, 0.01 for translation, 0.01 for scale, and 0.005 for adaptive kernel parameters, for a maximum of 200 iterations. We also have early stopping criteria, when $\sum(r(x; S_a, S_b^q, \mathbf{T}),\, x \forall A)/|A| \leq 0.0005$.

TABLE I

COMPARISON BETWEEN REG-NF AND NERF2NERF ON ONR DATASET.

		All Chairs Room					All Tables Room			Mix Room		
		c	dc	fc	fc-nop	mc	et	t	wt	dc	fc	t
$\Delta\mathbf{t} \downarrow$	FGR [29]	0.868	1.084	0.3715	0.485	0.152	0.368	0.819	0.219	1.351	0.226	0.636
	nerf2nerf [8]	0.278	0.291	0.127	0.117	**0.0007**	0.525	0.084	0.106	0.343	0.154	0.086
	Reg-NF (ours)	**0.04**	**0.035**	**0.018**	**0.014**	0.007	**0.398**	**0.044**	**0.022**	**0.292**	**0.029**	**0.009**
$\Delta\mathbf{R} \downarrow$	FGR [29]	1.130	1.669	1.020	1.449	0.1389	2.513	1.656	1.282	1.669	0.395	2.297
	nerf2nerf [8]	**0.041**	0.088	0.050	0.034	**0.002**	**2.396**	**0.002**	0.221	**0.050**	0.034	0.026
	Reg-NF (ours)	0.048	**0.039**	**0.031**	**0.020**	0.009	2.6	0.053	**0.025**	0.641	**0.030**	**0.012**
$\Delta s \downarrow$	FGR [29]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	nerf2nerf [8]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	Reg-NF (ours)	**0.019**	**0.007**	**0.007**	**0.060**	**0.003**	**0.019**	**0.004**	**0.006**	**0.021**	**0.009**	**0.005**

V. RESULTS

We first perform a quantitative analysis of Reg-NF, comparing it to nerf2nerf and demonstrate that we are outperforming them while not requiring manually annotated keypoints or an assumption that all objects are of the same scale as the scene. This is followed by two experiments demonstrating the robustness of Reg-NF to scale and the benefit of Reg-NF multi-view surface extraction. Finally, we demonstrate the benefits of Reg-NF for modelling imperfect scene NFs with known object NF replacement, and show how Reg-NF can enable object instance replacement for modelling alternative NF scenes with the same underlying object arrangements but different object NF models.

A. Comparison to nerf2nerf

We evaluate and compare the performance of Reg-NF and nerf2nerf [8], on our ONR dataset. We also compare against FGR for completeness [29]. Results shown in Table I are the average results across 10 iterations of the experiment to account for randomized factors such as RANSAC. In Table I we see that Reg-NF is typically at least an order of magnitude better than nerf2nerf in terms of $\Delta\mathbf{t}$ and is still generally superior in $\Delta\mathbf{R}$. We attribute the large increase of errors for nerf2nerf as being primarily due to the inherent scale differences between scene and database object models, for which nerf2nerf has no functionality to handle.

Focusing on Reg-NF, we note that failures can still occur, such as when we match object *dc* to scene *Mix Room* or *et* to scene *at-room*. In both these cases, we note the cause of failure being poor initialisation that proved inescapable for Reg-NF. A qualitative analysis of the Reg-NF registration can be seen in Fig 4 where coloured versions of NF object library models are substituted into the original scene NF at their poses calculated by Reg-NF.

B. Robustness to scale

We conducted an ablation study to investigate the performance of our approach under extreme scale differences between the two neural fields. We use Mix Room for our scene neural field and we attempt to align the *fc* object within that room to library *fc* models. For this test, we generated four additional *fc* models, each resized with scale factors $2, 0.5, 0.25, 0.1$. For these experiments, as convergence takes longer with large scale differences, we ran our optimisation procedure for 1000 iterations. Our results are shown in Table II. We observe that our approach can successfully align

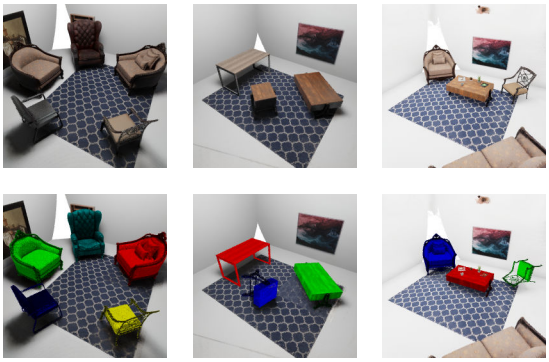


Fig. 4. Example of library replacement using Reg-NF for all objects in all scenes evaluated in Table I. Top: Original scene NF render. Bottom: Scene NF with library object NF substitutions render. Substitutions are based on Reg-NF outputs. Note, colours are added to object NFs during render to provide visual distinction between scene NF and object NFs.

TABLE II

REG-NF SCALE ANALYSIS STUDY ON MIX ROOM.

Scale Factor	$\Delta\mathbf{t} \downarrow$	$\Delta\mathbf{R} \downarrow$	Est. scale	True scale (GT)
2	1.40	1.40	2.39	2.5
1	0.03	0.03	1.24	1.25
0.5	0.001	0.002	0.62	0.62
0.25	0.30	1.42	0.62	0.31
0.1	0.17	2.58	0.34	0.13

the two object fields at a scale factor of 0.5. Interestingly, our learnt scale optimisation can successfully approximate the true scale factor between the scene and library models, even at a scale factor of 2. In extreme scale scenarios, we observe that the method has room for improvement.

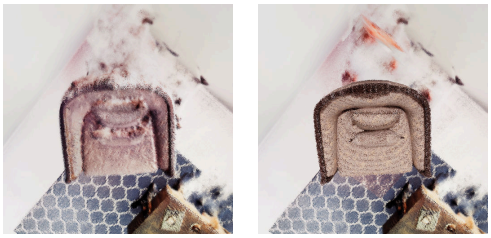
C. Effect of multi-view sample initialisation

The benefit of multi-view sampling during initialisation is most felt when an object has no distinguishing characteristics or is only partially seen from a single viewing angle taken from the training data. Using a single view this way introduces a high level of variability in object coverage. To show the impact of this, we perform experiments on the ac-room scene, comparing multi-view sampling to using a single challenging view extracted from the training data that observes the object. These views never see the front of chairs and often view only part of the chair, representing worst-case scenarios. The rest of the Reg-NF pipeline is kept consistent and quantitative results are shown in Table III. It is shown that a bad viewpoint with poor samples drastically reduces performance as no meaningful features could be extracted to enable effective

TABLE III

CHALLENGING SINGLE VIEW VS MULTI-VIEW TESTS

		c	dc	fc	fc-nop	mc
$\Delta\mathbf{t} \downarrow$	**Single**	0.307	0.674	0.378	0.681	0.147
	Multi. (ours)	**0.040**	**0.035**	**0.018**	**0.014**	**0.007**
$\Delta\mathbf{R} \downarrow$	**Single**	0.931	2.201	2.182	0.842	0.144
	Multi. (ours)	**0.048**	**0.039**	**0.031**	**0.020**	**0.009**
$\Delta s \downarrow$	**Single**	0.044	0.047	0.498	0.339	0.178
	Multi. (ours)	**0.019**	**0.007**	**0.007**	**0.060**	**0.003**



(a) Original scene (b) Library substitution

Fig. 5. Example of object completion via library replacement on a scene with low coverage. Original NF (a) is unable to correctly render the back of the object as it was not seen during training. (b) shows impact of library substitution from Reg-NF registration. Geometry of the object within the scene can be fully rendered from only partial initial view.

registration.

D. Substitution within imperfect scene models

We examine two types of "imperfect scene" to demonstrate practical applications for Reg-NF. The first considers when a robot may not be able to fully traverse a scene to get "full coverage" of an object for the scene's NF. For this we use the "short" trajectories of our mix-room scene. We can see in Fig. 5 that the scene NF trained from the short trajectory is not able to render the back of the chair clearly as it has no information about that model. Using Reg-NF, we register the object NF for the chair within the scene and can render a clear view of the back of the chair, fully understanding its geometry.

The second application setting considered for Reg-NF is operations with an under-trained scene model. While there are techniques available for speeding up NF training [4], in robotics, hardware limitations may lead to situations where a scene model cannot be fully trained before needing to be used. We therefore investigate the applicability of Reg-NF on highly under-trained scene NFs, trained with 300 steps rather than the 100,000 used in the main experiments. While being an extreme example, we show in Fig. 6 that an accurate alignment and object NF substitution is still feasible for some objects in this setting, enabling a clear render of the object even while the rest of the scene is under-defined.

E. Instance replacement

Finally, we demonstrate the benefits of using a library of pre-trained NF objects for creating new scenes. As all objects in the library are pre-defined and standardised, once Reg-NF derives the transform between a matched object NF and the scene NF, the known relative shapes/poses of objects within the NF library can be used to replace registered scene objects with any other object instance. We demonstrate this

(a) Under-trained scene (b) Library substitution

Fig. 6. Example of library replacement of object within under-trained scene NF. Original NF (a) has been under-trained but still needs to be used. (b) shows that if objects are detected, Reg-NF can provides needed transforms to replace the region of the detected object with a fully trained library model, providing clean object geometry.

(a) Original scene (b) Instance replacement

Fig. 7. Example of object instance replacement. After original chair in scene NF (a) has been properly registered against a matching object library NF instance via Reg-NF, it can be replaced by a different instance from the NF library (b).

within Fig. 7, showing *dc* being replaced with *mc*. This opens up potential use-cases in data-driven robotics where, after a scene has been turned into a neural field, objects within that scene can be changed to provide new data based on the layout of the original scene.

F. Limitations

The primary limitation of Reg-NF is the initialisation process. This is consistently the most extreme failure case in our work, when an initialisation is so poor that Reg-NF cannot find an optimal solution. This is a problem not faced by nerf2nerf due to the manual keypoint selection. We also acknowledge that current results have assumed a direct mapping from a scene object to a known object model of the same type, an assumption that cannot hold true for all real-world settings. This opens avenues of future work on template warping from a known library of objects.

VI. CONCLUSIONS

We present Reg-NF, a novel method for registration between neural field (NF) representations. Specifically, we estimate the 6DoF transform between objects found in a scene NF and object-centric NF counterparts stored in an NF object library, even when objects and scene have different scaling factors. We introduce a bi-directional registration loss and utilise the continuous nature of NF representations to align surfaces between objects and the scene. We analyse the effectiveness of Reg-NF and show its advantages for modelling objects within imperfect scene NFs and for enabling data-driven robotics research by offering editable scene NFs for robots to train in.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[2] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *CVPR*, 2021, pp. 7210–7219.

[3] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *ICCV*, 2021, pp. 5855–5864.

[4] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[5] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction," in *NeurIPS*, 2021, pp. 27 171–27 183.

[6] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *ICCV*, 2021, pp. 5589–5599.

[7] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *CVPR*, 2019, pp. 165–174.

[8] L. Goli, D. Rebain, S. Sabour, A. Garg, and A. Tagliasacchi, "nerf2nerf: Pairwise registration of neural radiance fields," in *ICRA*. IEEE, 2023, pp. 9354–9361.

[9] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[10] Q. Fu, Q. Xu, Y.-S. Ong, and W. Tao, "Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[11] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing nerf for editing via feature field distillation," in *NeurIPS*, vol. 35, 2022.

[12] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *CVPR*, 2022, pp. 5470–5479.

[13] J. Chibane, A. Mir, and G. Pons-Moll, "Neural unsigned distance fields for implicit function learning," in *NeurIPS*, 2020.

[14] C. Park, P. Moghadam, J. L. Williams, S. Kim, S. Sridharan, and C. Fookes, "Elasticity meets continuous-time: Map-centric dense 3D LiDAR SLAM," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 978–997, 2021.

[15] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.

[16] O. Kwon, J. Park, and S. Oh, "Renderable neural radiance map for visual navigation," in *CVPR*, 2023, pp. 9099–9108.

[17] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, "Loc-nerf: Monte carlo localization using neural radiance fields," in *ICRA*, 2023, pp. 4018–4025.

[18] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "Nerf-supervision: Learning dense object descriptors from neural radiance fields," in *ICRA*. IEEE, 2022, pp. 6496–6503.

[19] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3D neural scene representations for visuomotor control," in *Conference on Robot Learning*. PMLR, 2022, pp. 112–123.

[20] T. Weng, D. Held, F. Meier, and M. Mukadam, "Neural grasp distance fields for robot manipulation," in *ICRA*, 2023, pp. 1814–1821.

[21] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.

[22] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *CVPR*, 2022, pp. 12 786–12 796.

[23] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys, "Nicer-slam: Neural implicit scene encoding for rgb slam," *arXiv preprint arXiv:2302.03594*, 2023.

[24] Y. Chen and G. H. Lee, "DReg-NeRF: Deep Registration for Neural Radiance Fields," in *ICCV*, 2023.

[25] C. Peat, O. Batchelor, R. Green, and J. Atlas, "Zero nerf: Registration with zero overlap," *arXiv preprint arXiv:2211.12544*, 2022.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[27] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE international conference on robotics and automation*, 2009, pp. 3212–3217.

[28] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *PAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[29] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 766–782.

[30] Z. Yu, A. Chen, B. Antic, S. Peng, A. Bhattacharyya, M. Niemeyer, S. Tang, T. Sattler, and A. Geiger, "Sdfstudio: A unified framework for surface reconstruction," 2022.