Object Registration in Neural Fields

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

***Abstract*— Neural fields provide a continuous scene representation of 3D geometry and appearance in a way which has great promise for robotics applications. One functionality that unlocks unique use-cases for neural fields in robotics is object 6-DoF registration. In this paper, we provide an expanded analysis of the recent Reg-NF neural field registration method and its use-cases within a robotics context. We showcase the scenario of determining the 6-DoF pose of known objects within a scene using scene and object neural field models. We show how this may be used to better represent objects within imperfectly modelled scenes and generate new scenes by substituting object neural field models into the scene.**

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 [1]–[8]

Neural field (NF) 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. Multiple works have looked to utilising NFs for registration [9]–[11] but the first to look at direct registration of neural fields was nerf2nerf [9]. However, nerf2nerf does not suit the robotics domain as it relies on human-annotated keypoints for initialisation and assumes the scale of two neural fields are the same. Our recent work Reg-NF [12], overcomes these limitations to estimate the relative 6-DoF pose transformation between two objects of interest which are located in two different neural fields. Reg-NF 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. It builds upon the work on nerf2nerf, proposing a bidirectional registration loss, the use of multi-view sampling of the NF surface, and the use of SDFs [3], [5], [6], [13], [14] 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.

This paper is a companion to the original where we provide a condensed overview of our Reg-NF algorithm [12] and then extend the analysis of the original paper's results. These consider the use-case 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. This provides two particular benefits for NF object registration in robotics highlighted in Fig. 1. The first is object completion, using library substitution to improve the representation of scenes that have only partially observed objects. 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), generating new scene data for training in simulated NF environments.

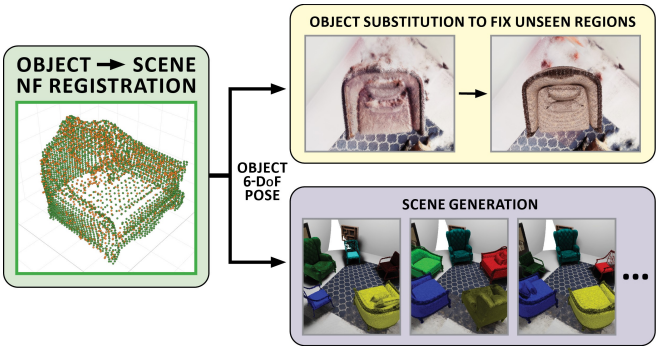


Fig. 1. The use-case for object registration in neural fields examined in this paper. After an object neural field (NF) is registered with its counterpart in a scene and 6-DoF pose has been attained, the library NF can be substituted into the scene to fix errors in the scene NF or new scenes can be generated using other object NFs and the known object pose in the scene.

II. Reg-NF

Reg-NF [12] provides a technique for aligning the surfaces of two different SDF NFs, by minimising the difference between their surfaces values. This can be used to provide the 6-DoF pose of an object within a scene given both are represented as neural fields. Here, we provide an overview of Reg-NF but invite readers to check the original paper [12] for full details.

For object registration, Reg-NF calculates the 6-DoF pose transformation $\mathbf{T}$ of a detected object from a pre-existing NF object library within a larger scene NF. This uses a differentiable optimisation function, initialised with an automated procedure. 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 library of object neural fields where $q \in \{1, ..., Q\}$. For our work, we utilise volumetric implicit surface fields, specifically NeuS [3] for all NFs in our work. These provide colour $c(\mathbf{x}, \mathbf{v})$ and signed surface distance $S(\mathbf{x})$ mapping functions derived from a shared backbone

Reg-NF 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

Accepted to the IEEE ICRA Workshop On Neural Fields In Robotics (RoboNerf) 2024

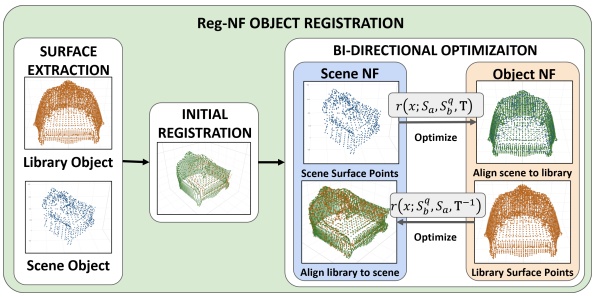


Fig. 2. Overview of the Reg-NF registration process [12]. Blue and orange denotes surface sample points from the scene and library NFs for a matched object respectively. Green points represent the target alignment during optimisation. After surface extraction and an initial registration estimate, bi-directional optimisation iterates till convergence. Final output is a 6-DoF transformation matrix between models.

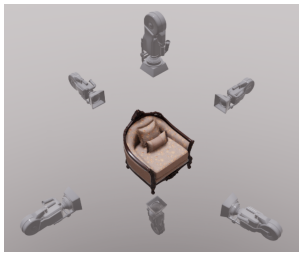


Fig. 3. Example of multiple views used for surface point extraction.

network with separate SDF and colour output heads when given a 3D point $\mathbf{x}$ and viewing direction $\mathbf{v}$. For full details on NeuS, please refer to [3]. An overview of Reg-NF is provided in Fig. 2.

A. Initial Registration

Reg-NF begins by establishing approximate correspondences between objects of interest within a and b^q. Assuming an object detection provides approximate location and classification of an object in a that matches to b^q, we calculate a set of N camera views pointed at the centroid of the detected object of interest as shown in Fig 3. We generate a grid pattern of rays travelling from each camera pose and return surface sample points P_a and P_b^q for our two respective neural fields. This multi-view sampling approach provides a clear geometric representation of the object for initialisation and leaves it less susceptible to poor surface sample initialisation from a single poor camera view.

Once surface samples are found for both NFs, we employ RANSAC [15] with Fast Point Feature Histogram (FPFH) [16] descriptors to estimate the correspondence between source and target, further refined via the point-to-point Iterative Closest Point (ICP) [17] method. Through this, we attain the initial six-degree-of-freedom (6-DoF) pose transformation $\hat{\mathbf{T}}$. Our initial registration takes ≈ 14 seconds on average.

B. Bi-directional Optimization

Reg-NF treats registration as an optimisation function that uses gradient decent to find the optimal pose transformation $\mathbf{T}$ between two surface fields initialised using $\hat{\mathbf{T}}$, and assuming an initial scale factor of $s = 1$ that minimizes a loss function L_s. Building on nerf2nerf [9], optimisation is performed over a discrete set of samples, with a robust kernel κ with learnt parameters p and α used to improve the robustness against

Fig. 4. Example images of object model images in ONR dataset.

outlier samples. However, Reg-NF improves optimisation by collecting surface samples points from both a and b^q (A and B^q), and performing a bidirectional optimisation over the model SDF outputs (S_a and S_b^q), as well as by including a regulariser designed to penalise when A and B^q deviate after transformation. Given this, the loss function is expressed as :

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

where registration residuals r are expressed as:

$$r(x; S_a, S_b^q, \mathbf{T}) = \|S_a - S_b^q \mathbf{T}\| \quad x \in A, \tag{2}$$

and

$$r(x; S_b^q, S_a, \mathbf{T}^{-1}) = \|S_b^q - S_a \mathbf{T}^{-1}\| \quad x \in B^q, \tag{3}$$

and L_r is our regulariser weighted with weight factor w expressed as:

$$L_r = \sum \frac{\min_{B^q} D^2_{A,B^q}}{|A|}, D^2_{A,B^q} = \|A - B^q \mathbf{T}\|^2, \tag{4}$$

where $|A|$ denotes the number of samples in A. Our loss function L_s is calculated using the current estimated 6-DoF pose transform $\mathbf{T}$ composed from translation components along all axes $[t_x, t_y, t_z]$, roll, pitch and yaw Euler angles $[r_r, r_p, r_y]$, and a scaling factor σ applied to all axes. Our loss function is then optimised over the learnt parameters: $(t_x, t_y, t_z, r_r, r_p, r_y, s, p, \alpha)$. Our optimization procedure takes ≈ 16 seconds on average.

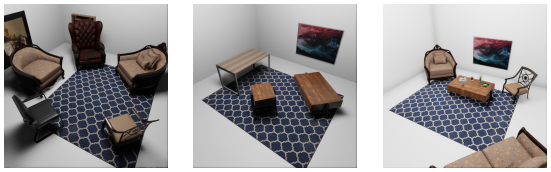
III. Experimental Design

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. ONR objects are outlined in Fig 4 There are three scenes collected: 1) containing all chair models stored

TABLE I

COMPARISON BETWEEN REG-NF OTHER REGISTRATION METHODS.

		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 [18]	0.868	1.084	0.3715	0.485	0.152	0.368	0.819	0.219	1.351	0.226	0.636
	nerf2nerf [9]	0.278	0.291	0.127	0.117	**0.0007**	0.525	0.084	0.106	0.343	0.154	0.086
	Reg-NF-init (ours)	0.300	0.322	0.093	0.110	0.125	0.383	0.210	0.144	0.447	0.138	0.100
	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 [18]	1.130	1.669	1.020	1.449	0.1389	2.513	1.656	1.282	1.669	0.395	2.297
	nerf2nerf [9]	**0.041**	0.088	0.050	0.034	**0.002**	**2.396**	**0.002**	0.221	**0.050**	0.034	0.026
	Reg-NF-init (ours)	0.213	0.196	0.044	0.074	0.032	2.566	0.235	0.140	0.195	0.099	0.016
	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 [18]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	nerf2nerf [9]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	Reg-NF-init (ours)	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**



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

Fig. 5. Scene models in ONR dataset.

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 are shown in Fig. 5.

Metrics: We follow the metrics in nerf2nerf [9] 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. We manually generated new human annotated keypoints for the initialisation procedure used in nerf2nerf.

Training models: All NF models were trained using the sdfstudio [19] implementation of NeuS which includes the proposal network from MipNeRF-360 [20] for training speed-up (neus-facto). For more details please refer to sdfstudio [19].

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.

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 $\sum(r(x; S_a, S_b^q, \mathbf{T}),\, x \forall A)/|A| \leq 0.0005$.

(a) nerf2nerf [9] (b) Reg-NF [12]

Fig. 6. Qualitative comparison of NF object registration methods for *dc* library model (red) in *ac-room* scene. Scale of *dc* in library starts much larger than within the scene causing nerf2nerf [9] to fail.

IV. RESULTS

We first perform a quantitative analysis of Reg-NF, comparing it to nerf2nerf, FGR, and the output of our initialisation step to 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 experiments showing 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 generating 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, nerf2nerf [9], FGR [18], and our Reg-NF initialisation on our ONR dataset. 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. An example of a nerf2nerf failure case is shown in Fig. 6

We also observe that despite the Reg-NF initialisation providing improved results over FGR, that the initialisation is rough and requires Reg-NF optimization to provide a close-fitting registration. Focusing on Reg-NF, we note that failures can still occur, such as when we match object *dc*

TABLE II

SINGLE VIEW VS MULTI-VIEW TESTS.

		c	dc	fc	fc-nop	mc	et	t	wt
Δt ↓	**Single**	0.307	0.674	0.378	0.681	0.147	0.449	0.632	0.733
	Multi. (ours)	**0.040**	**0.035**	**0.018**	**0.014**	**0.007**	**0.398**	**0.044**	**0.022**
ΔR ↓	**Single**	0.931	2.201	2.182	0.842	0.144	**1.562**	0.731	2.527
	Multi. (ours)	**0.048**	**0.039**	**0.031**	**0.020**	**0.009**	2.600	**0.053**	**0.025**
Δs ↓	**Single**	0.044	0.047	0.498	0.339	0.178	0.121	0.470	0.060
	Multi. (ours)	**0.019**	**0.007**	**0.007**	**0.060**	**0.003**	**0.019**	**0.004**	**0.006**

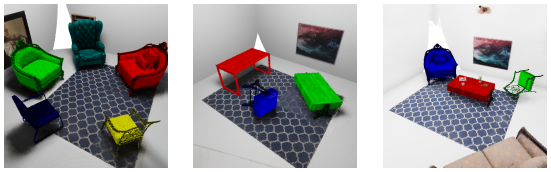


Fig. 7. Example of library substitution using Reg-NF for all objects shown in Fig 5. Substitutions are based on Reg-NF outputs. Colours are added to object NFs during render to provide visual distinction.

Fig. 8. Highly uninformative and/or partial views of models in ac-room and at-room from training data used for single-view experiments. In order: *et*, *t*, *wt*, *c*, *fc*, *dc*, *fc-nop* and *mc* objects.

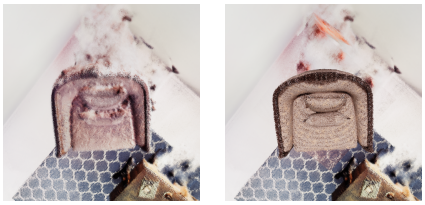
to scene *mix Room* or *et* to scene *at-room*. In both these cases, we note the cause of failure being poor initialisation that proved inescapable for Reg-NF, even if the error metrics of initialisation are lower than for some that were able to improve (e.g. *c* in *ac-room*). A qualitative analysis of the Reg-NF registration can be seen in Fig 7.

B. 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 introduces a high level of variability in object coverage. Using views shown in Fig 8 from neural field training data that only view part of the desired object for sampling the surface field, we test the worst-case scenario of single-view experiments. The rest of the Reg-NF pipeline is kept consistent and quantitative results are shown in Table II. This shows using bad viewpoints drastically reduces performance as no meaningful features could be extracted to enable effective registration.

C. Substitution within imperfect scene models

To demonstrate practical applications for substitution using Reg-NF, we consider when a robot may not be able to fully traverse a scene to get "full coverage" of an object for the scene's NF. We can see in Fig. 9 that a scene NF trained from a low-coverage trajectory cannot render the back of the chair clearly as it never saw that during training. Using



(a) Original scene (b) Library substitution

Fig. 9. Example of library substitution in a scene with low coverage. Original NF (a) cannot correctly render regions unseen during training. (b) library substitution after Reg-NF registration. Geometry of the object within the scene can be fully rendered from only partial initial view.

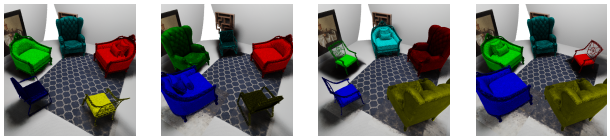


Fig. 10. Example of object instance replacement for generating new scenes. Original layout with coloured substitutions (left) followed by random replacement of chair models with others in the model library.

Reg-NF, we register the object NF for the imperfect chair within the scene and substitute it in the scene to render a clear view of the back of the chair.

D. Instance replacement for scene generation

Finally, we demonstrate the benefits of using a library of pre-trained NF objects for creating new scenes. 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 of the same class can be used to replace registered scene objects with any other object instance. We demonstrate this in Fig. 10 showing 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.

V. CONCLUSIONS

This paper provides an extended analysis of Reg-NF, a novel method for registration between neural field (NF) representations. Specifically, we examine the scenario where Reg-NF calculates the 6-DoF transform between objects found in a scene NF and object-centric NF counterparts stored in an NF object library. We analyse the effectiveness of Reg-NF, showcasing how it's bi-directional optimization improves upon initial registration, and how it's multi-view surface sampling provides robustness against naive single-view sampling. We demonstrate its advantages for modelling objects within imperfect scene NFs and for enabling data-driven robotics research by generating modified scene NFs for robots to train in.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

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

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

[12] S. Hausler, D. Hall, S. Mahendren, and P. Moghadam, "Reg-NF: Efficient registration of implicit surfaces within neural fields," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[13] 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.

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

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.