Appendix — NeRFDeformer: NeRF Transformation from a Single View via 3D Scene Flows

Supplementary Material

This supplementary material is structured as follows:

- We provide additional details regarding the NeRFDeformer in Sec. A.
- Additional information regarding baselines are summarized in Sec. B.
- Further experimental results are given in Sec. C.

A. NeRFDeformer Details

A.1. Optimization Details

As mentioned in the main paper, we do not store an independent learnable parameter R_i, t_i for each anchor point v_i, as this may be computationally demanding when the number of vertices/anchor points $|\mathcal{V}|$ is large. Instead, we only maintain learnable parameters for a subset of anchor points, *e.g.*, R_i^L, t_i^L. We then use interpolation to calculate R_i, t_i for the remaining anchor points via,

$$\mathcal{P}_i = \{(w_G(v_i, v_j^L), R_j^L) : v_j^L \in \mathcal{K}_\text{G}(v, \mathcal{V}^L)\}, \quad (1)$$

$$R_i = \text{quat\_average}(\mathcal{P}_i), \quad (2)$$

$$t_i = \sum_{j \in \mathcal{K}_\text{G}(v, \mathcal{V}^L)} w_G(v_i, v_j^L) t_j^L. \quad (3)$$

Here, $\mathcal{P}_i$ is the set of pairs of the rotation matrix R_i^L and their weights, $\mathcal{V}^L = \{v_i^L\}$ is the set of vertices on a coarser mesh $M^L = (\mathcal{V}^L, \mathcal{E}^L)$ obtained from mesh decimation of M^A and $\mathcal{V}^L \subset \mathcal{V}$. R_i is calculated using the classic quaternion averaging method [6] and t_i is obtained via a linear combination. We obtain the weight w_G as follows: $w_G(v_i, v_j^L) \propto$

$$1 - \frac{D_G(v_i, v_j^L, M^C)}{\max_{k \in \mathcal{K}_\text{G}(v, \mathcal{V}^L)} D_G(v_i, v_k^L)}. \quad (4)$$

Here, $D_G(v_i, v_k^L)$ calculates the geodesic distance on the denser mesh M^A between the anchor point v_i on M^A and the anchor point v_k^L on the coarser mesh M^L, *e.g.*, $\mathcal{V}^L \subset \mathcal{V}$. Also, $\mathcal{K}_\text{G}(v, \mathcal{V}^L)$ finds the K'-nearest neighbors ($K' = 6$) of v_i on the denser mesh M^A among vertices in $\mathcal{V}^L$ using the previous geodesic distance.

To reduce the computational needs, the standard as-rigid-as-possible loss L_ARAP is also formulated on the decimated mesh encouraging local rigidity for all vertices v_i^L:

$$L_\text{ARAP} = \frac{1}{|\mathcal{V}^L|} \sum_{i=1}^{|\mathcal{V}^L|} \sum_{j \in N(i)} \|(v_i^L + t_i^L - (v_j^L + t_j^L) - R_i^L(v_i^L - v_j^L)\|^2, \quad (5)$$

where $N(i)$ is v_i^L's neighborhood according to $\mathcal{E}^L$.

A.2. Correspondence Matching Details

First using the NeRF Φ we render images and corresponding depth maps depicting the original scene by sampling N random camera poses, i.e., we compute

$$\{(I_i^A, D_i^A) = \text{Render}(\Phi, E_i^A) : i \in \{1, \dots, N\}\}. \quad (6)$$

Here, E_i^A is a camera pose sampled approximately uniformly from a hemisphere over Φ.

We then apply ASpanFormer [2] over the image pair (I^B, I_i^A) depicting the transformed and original scene. Following ASpanFormer [2], we match pixels in a downsampled 100×100-resolution original image with pixels in the similarly downsampled transformed image. A threshold t_a ($t_a = 0.3$) is set to filter low confidence matches. Formally, using this procedure we obtain a set of 'transformed image'-'original image' pixel pairs $\hat{\mathcal{P}}_i = \{(\hat{p}_{i,j}^B, \hat{p}_{i,j}^A)\}$ for pair (I^B, I_i^A) referred to by index i, i.e.,

$$\hat{\mathcal{P}}_i = \text{ASpF}(I^B, I_i^A). \quad (7)$$

We apply ASpanFormer on multiple images as a single image's view is limited especially when non-rigid deformations relate the original and the transformed scene. Notably, correspondences are often noisy and may have conflicts. To address this, we use a two-step procedure to filter out wrong correspondences. As discussed in the main paper, the first step filters in 2D pixel space. The second step performs filtering in 3D space.

Pixel-space filtering: Let's use F_2D to refer to the 2D filtering function. It operates on all initial correspondence estimates $\hat{\mathcal{P}} = \cup_{i=1}^N \hat{\mathcal{P}}_i$ and yields

$$\hat{\mathcal{P}}' = F_\text{2D}(\hat{\mathcal{P}}). \quad (8)$$

Concretely, 2D filtering will retain only the best correspondence for each transformed pixel. 'Best' is measured by the size of the continuous patch of surrounding matching's transformed pixels with a confidence $> t_a$.

3D-space filtering: We found solely using 2D filtering to not be robust enough. A subsequent 3D filtering helps to further improve the results. For this, the above filtered 2D correspondences $\hat{\mathcal{P}}'$ are lifted to 3D space $\mathcal{P} = \{(p_i^B, p_i^A)\}$ given the depth maps for all transformed and original images and their camera poses. Here, p_i^B is a 3D transformed point by unprojection of a transformed pixel and p_i^A is a 3D original point located on the surface of Φ.

Table 1. Extra ablation results (ingp for Instant-NGP [8], ASpF for ASpanFormer [2]).

#	matching	original img source	Our flow	Filtering	New view synthesis			Geometric reconstruction		
					PSNR↑	SSIM ↑	LPIPS ↓	CD ↓	succ rate ↑	VmIoU ↑
4	ASpF	ingp	✓	2D + 3D	**25.9**$_{\pm 4.2}$	0.924$_{\pm 0.034}$	**0.061**$_{\pm 0.040}$	**1.46**$_{\pm 2.9}$	**0.903**	**0.666**$_{\pm 0.20}$
5	Lepard [5]	ingp	✓	2D + 3D	21.5$_{\pm 4.5}$	0.883$_{\pm 0.056}$	0.119$_{\pm 0.063}$	7.30$_{\pm 7.3}$	0.391	0.262$_{\pm 0.16}$
6-1	ASpF	gt	✓	2D + 3D	**25.9**$_{\pm 4.3}$	**0.925**$_{\pm 0.035}$	**0.061**$_{\pm 0.040}$	2.68$_{\pm 11}$	0.885	0.646$_{\pm 0.21}$
6-2	ASpF	vanilla NeRF	✓	2D + 3D	25.0$_{\pm 4.2}$	0.911$_{\pm 0.037}$	0.080$_{\pm 0.045}$	2.86$_{\pm 4.6}$	0.782	0.585$_{\pm 0.23}$

Table 2. Geometric reconstruction under noisy camera pose.

s (degree)	CD ↓	CD (success) ↓	succ rate ↑	VmIoU ↑
0	1.46$_{\pm 2.9}$	0.62$_{\pm 0.79}$	0.903	0.666$_{\pm 0.20}$
3	2.80$_{\pm 11}$	0.81$_{\pm 0.84}$	0.884	0.646$_{\pm 0.21}$
5	3.02$_{\pm 11}$	1.00$_{\pm 0.89}$	0.876	0.642$_{\pm 0.21}$
10	4.06$_{\pm 12}$	1.56$_{\pm 0.80}$	0.779	0.649$_{\pm 0.21}$

Let's use F_{3D} to refer to filtering in 3D space, which is implemented in an iterative way, i.e.,

$$\mathcal{P}^{k+1} = F_{3D}(\mathcal{P}^k) \text{ for } 0 \leq k < K_{3D}. \tag{9}$$

We let $\mathcal{P}^0 = \mathcal{P}$. Here, $\mathcal{P}^k = \{p_i^k\}$ is the 3D correspondence pair set after the k-th 3D filtering. The intuition of 3D filtering is that if several pairs have adjacent original points, they are also probably adjacent on the original NeRF's surface, and should not have drastically different transformed correspondence. Otherwise there are false positives. More specifically, for each 3D corrspondence pair $p_i^k = (p_i^{B,k}, p_i^{A,k})$, we first select all adjacent pairs p_j^k with a similar original point $p_j^{A,k}$ that $\|p_i^{A,k} - p_j^{A,k}\| < r (r = 0.01)$ to form a pair index set $\mathcal{I}_i^k$. Then calculate the average transformed point $p_i'^{B,k}$ of all transformed points in the former set:

$$p_i'^{B,k} = \frac{1}{|\mathcal{I}_i^k|} \sum_{j \in \mathcal{I}_i^k} p_j^{B,k}, \quad \text{with} \tag{10}$$

$$\mathcal{I}_i^k = \{j, \forall j \text{ s.t. } \|p_i^{A,k} - p_j^{A,k}\| < r\}. \tag{11}$$

After this, if the transformed point of pair i is not very different from the above average, *e.g.*, $\|p_i^{B,k} - p_i'^{B,k}\| < r'$ where we have $r' = 0.02$. The pair i is kept in $p^{A,k+1}$ and $p^{B,k+1}$, otherwise, it's removed.

B. Baseline Details

DreamGaussian we run their released code. Importantly, to resolve scale ambiguity between the mesh extracted from DreamGaussian and the ground truth one, we fit a scale and translation factor to minimize the Chamfer distance.

NeRF finetuning we finetune the original NeRF for an additional 2k iterations via the default NeRF reconstruction loss given only the transformed view.

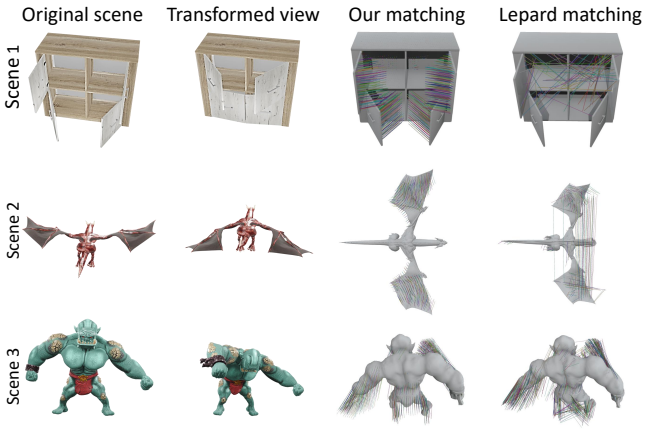

Figure 1. Corrspondences matching of Ours and Lepard [5]. We observe that Lepard produces too many wrong correspondences while ours are cleaner.

SINE we re-implementated their method as best as we could since at the time of this submission there was no publicly available code and training/testing data. We carefully follow the paper while changing a few hyperparameters to improve their results further on our dataset. The "shape prior" part of SINE follows our mesh transformation. No details of how SINE transformed the mesh are provided in the paper, *e.g.*, mesh resolutions and losses. The MLP for both 3D flow directions follows the SINE paper design with 1 hidden layer and 128 hidden size using ReLU activation. We also use positional encoding with 4 frequencies. The loss terms L_{gp}, L_{gr}, and L_{cycle} follow the SINE paper. We also use their coefficients for losses, except increasing the L_{gp} coefficient from 0.03 to 1 as we observe the original one to struggle to model the flow of the transformation.

C. Extended Experiments

C.1. Extended Ablations

The quantitative results of our extra ablation study are provided in Tab. 1 and Tab. 2.

Point cloud matching. We first analyze the performance of another choice of correspondence matching: Using point cloud matching [1, 3–5] to find 3D correspondences di-

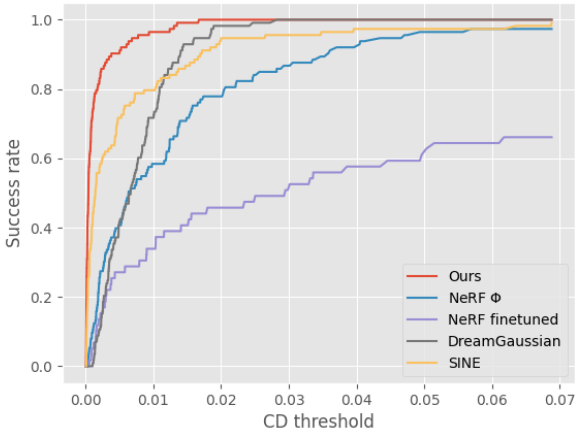

Figure 2. ROC curves of Chamfer distance for different methods.

rectly. To test this, we first unproject the transformed depth map into a point cloud. We then sample points on the surface of M^A, the mesh of the original scene. Finally we apply the current state-of-the-art point cloud matching method Lepard [5] on both point clouds. Comparing our design row 4 and Lepard's in row 5 in Tab. 1. We observe that the latter obtains worse results both in geometry reconstruction and new view synthesis. We hypothesize that this is because Lepard ignores the rich information from RGB images, making capturing of the non-rigid transformations observed in our task challenging. More correspondence matching results of both are illustrated in Fig. 1.

NeRF quality. We also analyze the influence of the NeRF quality in our method. Note that we are using Instant-NGP as the NeRF representation in the main paper. We test two baselines here. Row 6-1 replaces the original images rendered from the NeRF for correspondence matching with ground truth images. Row 6-2 replaces the Instant-NGP of the original scene with a vanilla NeRF [7] trained from fewer and lower resolution images (200 images, 800×800 resolution). We observed that row 4 and 6-1 have similar performance. Although a lower NeRF quality degrades our method's result (see row 6-2), the method still achieves decent results, better than all baselines.

Camera quality To assess the impact of camera pose quality, we perturbed the camera by adding to pitch and yaw a random number drawn uniformly between $[-s, s]$ (see Tab. 2). Although incorrect camera pose degrades results (as expected), our method outperforms the other baselines despite their use of perfect camera pose.

C.2. Extended Results

Extra geometry reconstruction evaluation. In the main paper we only put the success rate of chamfer distance (CD) on the threshold $= 0.004$. To explain the geometry reconstruction quality of all methods better, we also draw ROC curves of CD in Fig. 2. We observe that our method's reconstruction is better than the one obtained from baselines on any thresholds.

Failure case. The method can't recover if too many correspondences are wrong as illustrated in Fig. 3.

Extra visualizations. Furthermore, we extend the qualitative figure of the main paper in Figs. 4 to 6, which show more results on scenes of our dataset constructed from Objaverse assets. For a better understanding of the scene transformation, we also visualize the view of the original scene with the same camera pose as the transformed view's.

(a) Original scene (b) Transformed view (c) Ours matching

Figure 3. A failure case of our correspondence matching. (a,b) depict the original and transformed scene. (c) is our matching result. See that points on one leg are matched to another. Legs all look similar and correspondence matching method fails to distinguish them.

References

[1] Aljaz Bozic, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural non-rigid tracking. *Advances in Neural Information Processing Systems*, 33: 18727–18737, 2020. 2

[2] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *European Conference on Computer Vision*, pages 20–36. Springer, 2022. 1, 2

[3] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the european conference on computer vision (ECCV)*, pages 230–246, 2018. 2

[4] Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J Guibas. Non-rigid registration under isometric deformations. In *Computer Graphics Forum*, pages 1449–1457. Wiley Online Library, 2008.

[5] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5554–5564, 2022. 2, 3

[6] F Landis Markley, Yang Cheng, John L Crassidis, and Yaakov Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007. 1

[7] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:

Figure 4. Original view in the same camera pose as the transformed view. Ground truth, DreamGaussian, SINE, and our method.

Figure 5. Original view in the same camera pose as the transformed view. Ground truth, DreamGaussian, SINE, and our method.

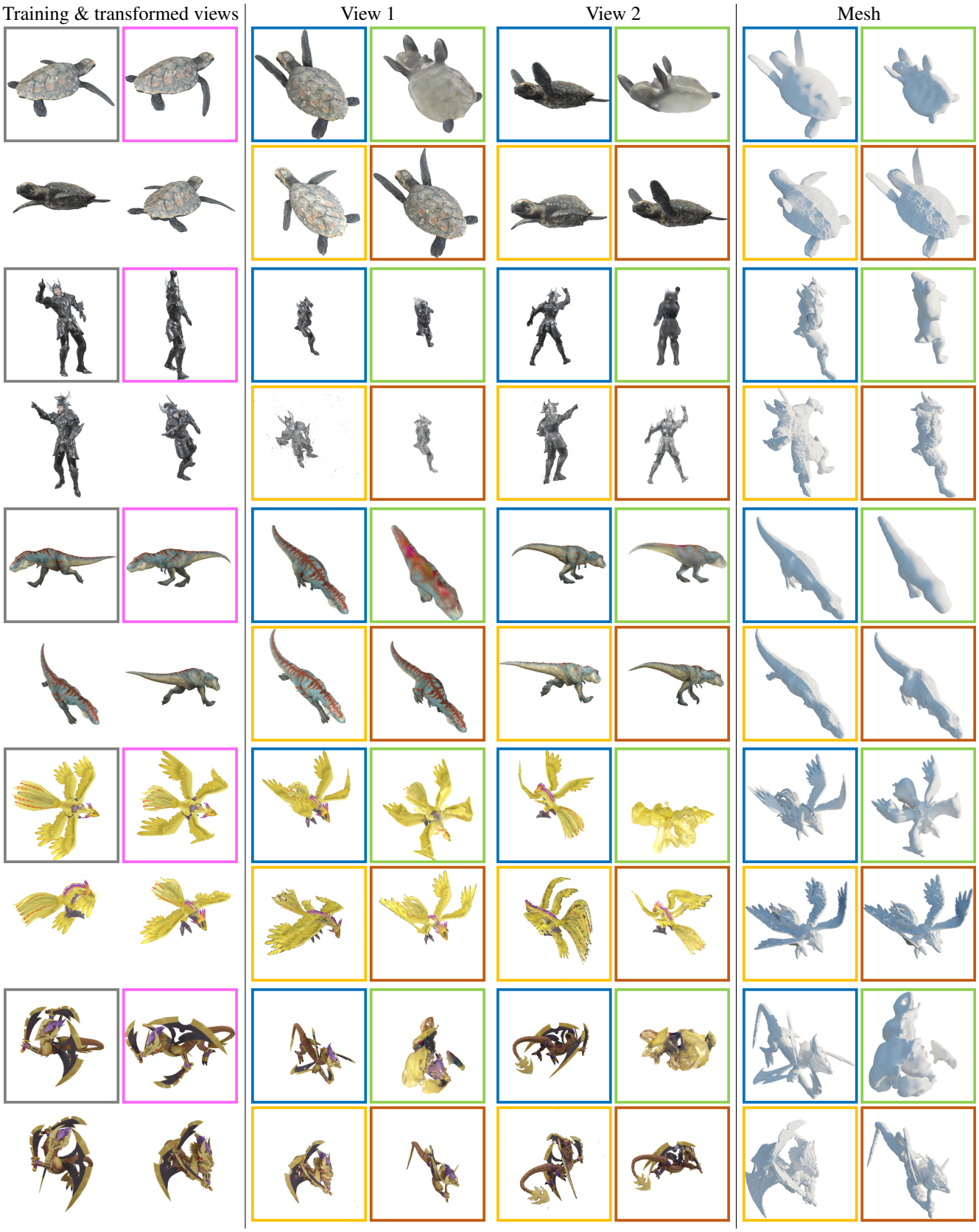

Figure 6. Original view in the same camera pose as the transformed view. Ground truth, DreamGaussian, SINE, and our method.

Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[8] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2