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

Zhenggang Tang,[1] Zhongzheng Ren,[1] Xiaoming Zhao,[1] Bowen Wen,[2] Jonathan Tremblay,[2] Stan Birchfield,[2] Alexander Schwing[1]

[1]University of Illinois Urbana-Champaign: {ztang, zr5, xz23, aschwing}@illinois.edu

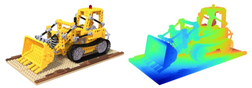
[2]NVIDIA: {bowenw, jtremblay, sbirchfield}@nvidia.com

Task Definition

Goal: Automatically transforming a NeRF given a **single** observation I^B showing a **non-rigidly** transformed version (B) of the original NeRF-captured scene (A)

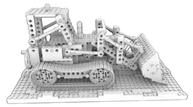
Input:

Original NeRF trained on multiview scene



Single RGB-D from transformed scene

Output:



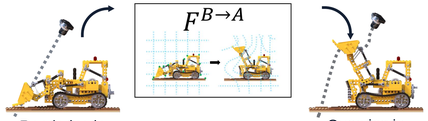
Exported mesh of the transformed scene

Novel views of the transformed scene

Method Overview

We define two **flows**: original (A) to transformed (B) scene **forward** flow ($F^{A\to B}$) and **backward** flow ($F^{B\to A}$)

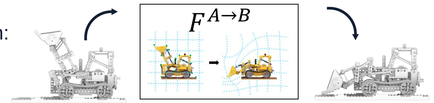
New view synthesis:



Rendering in transformed space B

Querying in original NeRF A

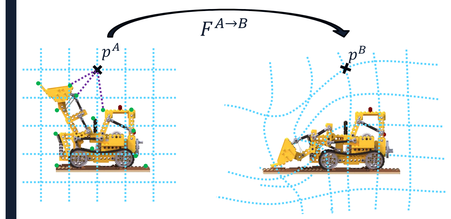
Geometry reconstruction:



Original mesh M^A

Transformed mesh M^B

Flow Definitions: Embedded Deformation Graph



Original space A Transformed space B

Rigid transformations ξ_i anchored at mesh vertices $\mathcal{V} = \{v_i\}$.
$\xi_i(p^A) = R_i(p^A - v_i) + v_i + t_i$,
R_i and t_i are learnable parameters
p^A is a query point

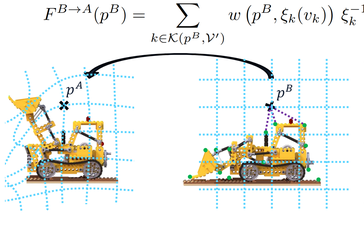
Final transformation: a blending of ξ_i

$$F^{A\to B}(p^A) = \sum_{k\in\mathcal{K}(p^A,\mathcal{V})} w(p^A, v_k)\,\xi_k(p^A),$$

Weight $w(p^A, v)$ via K-nearest vertex neighbor search $\mathcal{K}$

$$w(p^A, v) \propto \left(1 - \frac{\|v - p^A\|}{\max_{k\in\mathcal{K}(p^A,\mathcal{V})}\|v_k - p^A\|}\right)$$

$$F^{B\to A}(p^B) = \sum_{k\in\mathcal{K}(p^B,\mathcal{V}')} w\left(p^B, \xi_k(v_k)\right)\,\xi_k^{-1}(p^B)$$



Original space A Transformed space B

- No need to optimize again
- Use ξ_i^{-1} as the naïve inverse of ξ_i
- $\mathcal{K}$ is querying in the transformed vertices set $\mathcal{V}' = \{\xi_i(v_i)\}$
- Only cyclic on near surface areas → more flexible flows

Loss

We optimize the parameters of rigid transformations ξ_i via

$$L_{\text{DG}} = L_{\text{ARAP}} + \alpha L_{\text{Con}}$$

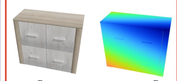
As-rigid-as-possible regularization using vertex connectivity

Translation penalty from corresponding points $I = \{(v_i^A, v_i^B)\}$

$$L_{\text{Con}} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\|t_i + v_i^A - v_i^B\|^2$$

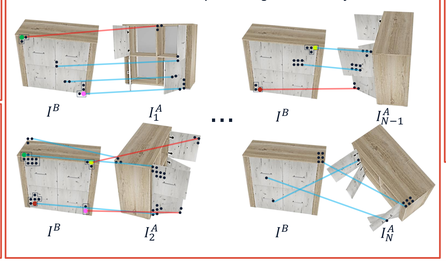
Robust NeRF-based Correspondence Matching

Transformed view

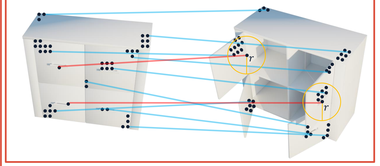


Rendering of original images I_i^A

1. Pixel-space filtering: confidence measured via pixel neighbor density size



2. 3D-space filtering:



- Render images I_i^A of original scene A via NeRF
- 2D dense correspondence matching between I^B and I_i^A; filter out less confident pairs
- Lift pixel correspondences to 3D space; filter out pairs behaving differently from nearby clusters

Quantitative Results

Methods	New view synthesis			Geometric reconstruction			
	PSNR ↑	SSIM ↑	LPIPS ↓	CD ↓	CD (success) ↓	succ rate ↑	VmIoU ↑
Zero123-XL [5]	14.1$_{\pm 3.9}$	0.799$_{\pm 0.071}$	0.265$_{\pm 0.076}$	/	/	/	/
DreamGaussian [40]	19.8$_{\pm 4.2}$	0.868$_{\pm 0.057}$	0.149$_{\pm 0.067}$	7.36$_{\pm 5.1}$	2.46$_{\pm 0.84}$	0.336	0.306$_{\pm 0.18}$
NeRF Φ	21.3$_{\pm 3.6}$	0.876$_{\pm 0.059}$	0.125$_{\pm 0.061}$	13.2$_{\pm 16}$	1.72$_{\pm 0.95}$	0.372	0.315$_{\pm 0.23}$
NeRF finetuned	21.6$_{\pm 3.5}$	0.826$_{\pm 0.096}$	0.198$_{\pm 0.100}$	228$_{\pm 270}$	1.85$_{\pm 1.10}$	0.195	0.312$_{\pm 0.25}$
SINE [1]*	22.1$_{\pm 3.8}$	0.883$_{\pm 0.052}$	0.115$_{\pm 0.053}$	6.40$_{\pm 13}$	1.85$_{\pm 1.10}$	0.637	0.515$_{\pm 0.25}$
Ours	**25.9**$_{\pm 4.2}$	**0.924**$_{\pm 0.034}$	**0.061**$_{\pm 0.040}$	**1.46**$_{\pm 2.9}$	**0.62**$_{\pm 0.79}$	**0.903**	**0.666**$_{\pm 0.20}$

Real world data:

Original & transformed scene

Our rendering & geometry reconstruction

Qualitative Results

Objaverse data:

Transformed views, GT, DreamGaussian, SINE, Ours