# Multiview Regenerative Morphing with Dual Flows

Chih-Jung Tsai[1], Cheng Sun[1,2], and Hwann-Tzong Chen[1,3]

[1] National Tsing Hua University
[2] ASUS AICS Department
[3] Aeolus Robotics

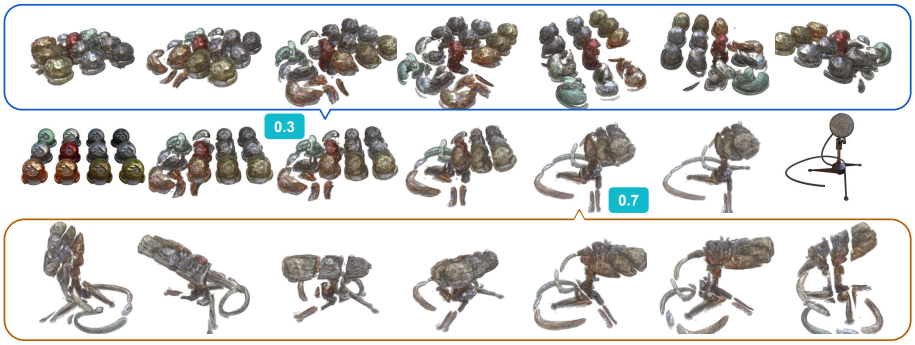

**Fig. 1.** Multiview Regenerative Morphing.
The middle row shows an example of *Multiview Regenerative Morphing* from the source (left) to the target (right). The top row shows multiview rendering when the blending weight is 0.3. The bottom row shows multiview rendering with blending weight 0.7.

**Abstract.** This paper aims to address a new task of image morphing under a multiview setting, which takes two sets of multiview images as the input and generates intermediate renderings that not only exhibit smooth transitions between the two input sets but also ensure visual consistency across different views at any transition state. To achieve this goal, we propose a novel approach called Multiview Regenerative Morphing that formulates the morphing process as an optimization to solve for rigid transformation and optimal-transport interpolation. Given the multiview input images of the source and target scenes, we first learn a volumetric representation that models the geometry and appearance for each scene to enable the rendering of novel views. Then, the morphing between the two scenes is obtained by solving optimal transport between the two volumetric representations in Wasserstein metrics. Our approach does not rely on user-specified correspondences or 2D/3D input meshes, and we do not assume any predefined categories of the source and target scenes. The proposed view-consistent interpolation scheme directly works on multiview images to yield a novel and visually plausible effect of multiview free-form morphing. Code: https://github.com/jimtsai23/MorphFlow

## 1 Introduction

Image morphing is an appealing visual effect that transforms one image into another with coherent intermediate results showing smooth transitions. It has wide applications in visualization, special visual effect, and virtual reality. Conventional morphing methods consist of three steps: *i*) acquire user-specified landmark-based or dense correspondences, *ii*) warp each image into an intermediate layout based on the aligned correspondences, and *iii*) blend the two aligned images with the respective weights. The need of user-specified correspondences is unfavorable and sometimes even impossible when image contents are very dissimilar. On the other hand, as observed in [47], simple warping functions may cause unnatural appearances when characterizing complex deformations. Later approaches like *regenerative morphing* [48], and more recently the GAN-based methods such as [29,30], are able to relax the requirement of explicitly specified correspondences and produce intriguing effects of image-to-image warping and interpolation.

In this work, we build upon the success of prior techniques and aim to solve a more challenging task of image morphing to render multiview morphs between two structurally unaligned and visually unrelated scenes, as shown in Fig. 1. More precisely, the new task we seek to address can be described as follows: Consider two sets of images, each taken from a scene of arbitrary categories under various viewpoints. The goal of our task is to build a model that has the capability of producing multiview morphs, such that, *i*) at any chosen viewpoint, it can generate a sequence of transitions as in standard image morphing, while *ii*) at any given transition moment, it can present multiview renderings of the intermediate morphing scene. To achieve this goal, we propose to learn a model comprising volumetric scene representations for rendering morphs. Each scene is represented by a 4D volume, where each voxel contains RGB and alpha (opacity) values. The representation is learned in a coarse-to-fine manner. First, we use a coarse voxel grid to locate the probable occupancy of the scene, and then we use a finer voxel grid to optimize for the details. Instead of using implicit functions or any kind of neural network, we adopt an explicit differentiable volume rendering scheme to reduce computation time.

Suppose that we have derived the aforementioned representations from the multiview images of the source and target scenes. Now, to proceed with the morphing task, we need to fuse the two representations into an intermediate one that can be used to render novel views of the morphs for any given transition moment. Without relying on predefined correspondences, we adopt an optimal transport mechanism that models intermediate transitional representations as interpolations of the original two representations. The interpolations resemble a mixture of scaffolds of the two scenes for querying and blending the volumetric representations. As free-form regenerative morphing may result in shattered structures during transitions, we regularize the morphing process by enforcing a rigid transformation to avoid generating fragmented morphs. With the interpolated representation and rigid transformation, our method can render a sequence of coherent morphs of the two category-independent input scenes. The rendered morphs can be displayed from different viewpoints at different transition moments.

We summarize the main ideas of this work as follows:

1. This paper presents an optimization-based method that tackles a new task of multiview regenerative morphing as illustrated in Fig. 1. The proposed method takes multiview images as the input; no 2D or 3D meshes are needed.
2. Our approach does not assume the categories of and the affinities between the source and target images, nor does it require any predefined correspondences between them.
3. Our approach adopts the mechanism of optimal transport to get an interpolated volume for rendering transitional multiview morphs. We also include a rigid transformation in the morphing process to favor 'structure-preserving' morphs when possible.
4. Our method is efficient in learning and rendering. It can learn a morphing renderer from scratch (directly from the input images) in 30 minutes. For morphing and rendering, the learned renderer can generate one novel-view morph per second.

## 2 Related Work

**Image morphing.** Image morphing aims at transforming a source image to a target image smoothly and with natural-looking in-between results. Traditional approaches [33,56] use image warping and color interpolation with predefined dense correspondence. In particular, [2] enforces the transition to be as rigid as possible, and [47] considers camera viewpoint to prevent distortion. Patch-based methods [10,48] are later proposed to synthesize in-between images using source and target patches under temporal coherence constraints. Recently, Generative Adversarial Networks (GAN) has shown impressive image generation results by learning a projection from latent space to image space. Image morphing can then be achieved by simple linear blending in the GAN latent space [1,19,44,58]. Simon and Aberdam [50] further propose to solve the Wasserstein barycenter problem constrained on GAN latent space to achieve smooth transitions and natural-looking in-between results. Optimal transport has also been used to produce morphing between simple 2D geometries [4–6, 51]. However, the morphs lack textures as in nature images. Our method is different from the above morphing methods in that we generate 3D representations and render view-consistent morphs in novel views. Perhaps the most similar to us are the multilevel free-form deformation morphing techniques described in [56]. While they still depend on 3D primitives and human-labeled correspondence, our morphing technique is fully automatic and unsupervised.

**Volume renderer from multiview images.** Reconstructing a volumetric scene representation that supports novel-view synthesis from a set of images is a long-standing task with steady progress [15,17]. NeRF [41] has recently revolutionized this task by incorporating the coordinate-based multilayer perceptrons

(MLP) to represent each spatial point's color and volume density implicitly. The MLP model is trained to minimize the photometric loss on the observed views with differentiable volume rendering. Many follow-up works of NeRF are proposed to achieve better qualities on background [63], surface [43], multi-resolution [3], imperfect input poses [27, 36, 40], fewer input views [7, 53, 62], and dynamic scene [20, 35, 39, 59]. Despite the high quality and flexibility, NeRF still has a disadvantage of its lengthy training and rendering run-time. To improve rendering speed, many methods are proposed to convert the trained implicit MLP representations to explicit voxel-grid or hybrid representations [21, 23, 55, 61]. The improvement on training run-time relies on cross-scene pre-training [7, 53, 62] or external depth [11, 38]. Our morphing algorithm is agnostic to the underlying scene representation reconstruction techniques. For ease of use, our representation explicitly models the scene with voxels, similar to DirectVoxGO [52]. We reconstruct the source and target scene representations from their respective image sets, and use the learned volumetric representations for morphing.

**Shape interpolation.** Given two (or more) shapes, shape interpolation aims at generating their in-between shape specifying by a composition percentage, which enables a smooth deformation from one shape to the other. Traditional methods try to recover the shape space manifold [24,25,31,54], and then the shape interpolation becomes a geodesic-path searching problem on the manifold. As shape manifold recovering is challenging, some recent approaches [12, 13] directly find the deformation field from source to target shapes but with isometric (zero-divergence, constant volume) assumption. NeuroMorph [14] uses neural networks to predict the correspondences and the deformation field, which works well even for non-isometric pairs. Generative neural networks have recently achieved good results in 3D by building shape latent spaces and using deep decoders to map latent codes to voxels [57], signed distance fields [28, 45], point clouds [9, 34, 49], or meshes [8]; shape interpolation is then achieved by linearly blending the latent codes of the two shapes. These shape-interpolation techniques typically take 3D as input (*e.g.*, mesh), and GAN-based methods further require large datasets, while our method only needs two sets of images that capture the source and the target shapes. Janati *et al.* and Solomon *et al.* have considered interpolations between shapes as Wasserstein barycenters [26, 51]. However, their computations do not include morph in appearances. To facilitate learning on more complicated shapes, we take a different strategy and morph the scenes with the Wasserstein flow [18]. We further regularize the flow with rigidity constraints, and use these local and global dual flows to achieve multiview regenerative morphing.

## 3 Overview

Consider the images $\mathcal{I}^{\mathcal{S}}$ and $\mathcal{I}^{\mathcal{T}}$ collected from the source and target scenes with camera poses $\zeta^{\mathcal{S}}$ and $\zeta^{\mathcal{T}}$. Our method can generate morphed images between $\mathcal{I}^{\mathcal{S}}$ and $\mathcal{I}^{\mathcal{T}}$ given arbitrary weights $t$ and viewing angles $\theta, \phi$. The method has two phases. In the first phase, we establish volumetric representations for each scene

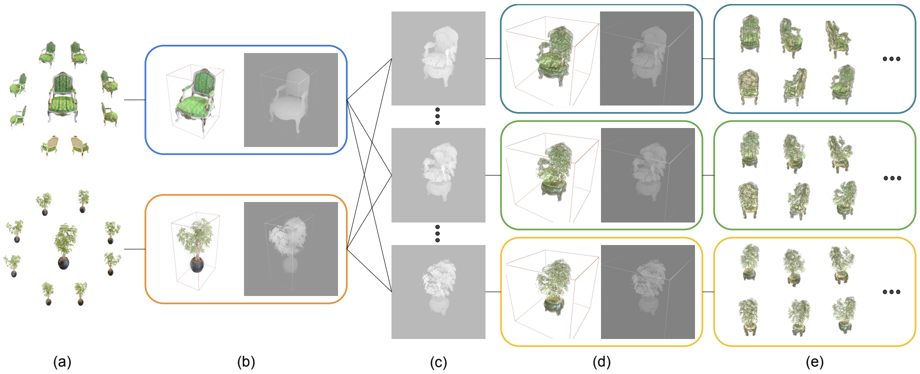

**Fig. 2.** An overview of our method. (a) The input multiview images of the source and target scenes. (b) The volumetric representation of each scene comprises color and opacity information. (c) A sequence of the morphed point sets with different blending weights. They are interpolations between source and target scene in Wasserstein metrics. (d) The volumetric representations generated by morphing. (e) Examples of multiview morphing rendered at arbitrary viewing directions. It can be seen that the rendering results are view-consistent—at any moment, the intermediate morph can be viewed as an actual scene and does not exhibit any conflicts across views.

with the purpose of generating view-consistent morphs from unaligned images. The representation comprises opacity and color. In the second phase, we use optimal transport to generate morphs between the derived volumetric representations from the first phase. The morphing process is controlled by rigid transformation (RT) flow and optimal transport (OT) flow. RT flow preserves the impression of the source scene during morphing, preventing shattered generation. OT flow deforms the source scene into the target without the need of correspondences. By applying the two flows, we obtain a morphed representation and render view-consistent morphs in any views. Fig. 2 illustrates the pipeline of our method.

Formally, we use a volume $\mathcal{V}$ to model a scene by mapping a 3D position $\mathbf{x} = (x, y, z)$ to its corresponding opacity $\alpha$ and color $\mathbf{c} = (r, g, b)$ as

$$\mathcal{V} : \mathbb{R}^3 \to \mathbb{R}^4, \quad \mathcal{V}(\mathbf{x}) = (\mathcal{V}_\alpha(\mathbf{x}), \mathcal{V}_\mathbf{c}(\mathbf{x})) \ , \tag{1}$$

where $\mathcal{V}_\alpha(\mathbf{x})$ retrieves the opacity $\alpha$ and $\mathcal{V}_\mathbf{c}(\mathbf{x})$ yields the color $\mathbf{c}$. Based on Eq. (1) we build volumes $\mathcal{V}^\mathcal{S}$ and $\mathcal{V}^\mathcal{T}$ for representing the source and target scenes. The opacity can be used to filter out negligible voxels. The balance between granularity and efficiency is controlled by a threshold $\delta_\alpha$, *i.e.*, voxel $v_i$ in $\mathcal{V}_\alpha$ is collected if $\alpha_i > \delta_\alpha$. The collection of points and opacity values serves as a shape representation, which can be expressed as a weighted point set $\mathcal{P} = \{(\omega_i, \mathbf{x}_i)\}_{i=1}^N$, where $\omega_i = \alpha_i / \sum_j^N \alpha_j$, associating each point with a weight derived from the opacity by normalization. In this way, we create a source shape $\mathcal{S}$ from the source volume $\mathcal{V}^\mathcal{S}$ and a target shape $\mathcal{T}$ from the target volume $\mathcal{V}^\mathcal{T}$, where both are in the form of a weighted point set as described above. While we transform the opacity volume into the source shape $\mathcal{V}^\mathcal{S}$, for each point $i$ in the source set, its color $\mathbf{c}_i$ can be gathered from the color volume $\mathcal{V}_c^{\mathcal{S}}$. The point colors are preserved for blending the final appearances.

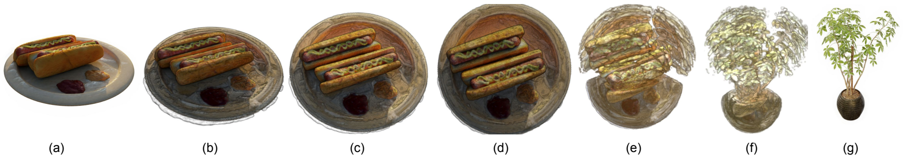

**Fig. 3.** To visualize the different effects of the rigid transformation (RT) flow $\mathbf{f}$ and the optimal transport (OT) flow $\mathbf{g}$, we deliberately apply them one after the other. (a–d): we apply only $\mathbf{f}$, which aligns the source's pose with the target's without changing the shape. (d–g): we apply only $\mathbf{g}$, which deforms from aligned source shape to target shape. In our method, the two flows are jointly applied during the entire morphing process.

To morph between the source shape $\mathcal{S}$ and the target shape $\mathcal{T}$, we develop a two-step algorithm using the *2–Wasserstein distance*. In the first step, we seek a rigid transformation $\hat{\Psi}$ that minimizes the distance between the source and target shapes in the Wasserstein space. We solve for $\hat{\Psi}$ using gradient descent on unbiased Sinkhorn divergence SD. A more detailed algorithm is described in Sec. 5.1. The second step finds an interpolation between the rigidly transformed source shape $\hat{\Psi}(\mathcal{S})$ and the target shape $\mathcal{T}$. We obtain the interpolation via a gradient-guided step on SD [18,22], which is interpreted as a displacement vector from the source shape to the target shape. As a result, two flows are created, namely, the RT flow $\mathbf{f}$ and the OT flow $\mathbf{g}$, by comparing the transformation $\hat{\Psi}(\mathcal{S})$ to the source shape and the target shape to the transformed source. A morphed shape can then be generated by

$$\mathcal{M}_t \leftarrow \mathcal{S} + \mathbf{f}(t) + \mathbf{g}(t) \,, \tag{2}$$

where the flow parameter $t \in [0,1]$ controls the progression of $\mathbf{f}$ and $\mathbf{g}$ at each transition moment $t$ to create the expected morphs. Fig. 3 visualizes the effects of the two flows.

We use the morphed shape $\mathcal{M}_t$ to query color volume of target $\mathcal{V}_{\mathbf{c}}^{\mathcal{T}}$ and generate blended colors $\mathcal{V}_{\mathbf{c}}^{\mathcal{M}_t}$ along with colors of source points. The morphed shape is voxelized to form $\mathcal{V}^{\mathcal{M}_t} = \left(\mathcal{V}_{\alpha}^{\mathcal{M}_t}, \mathcal{V}_{\mathbf{c}}^{\mathcal{M}_t}\right)$ so that we can render view-consistent morphing images using $\mathcal{V}^{\mathcal{M}_t}$ under arbitrary viewing directions.

In short, the proposed Multiview Regenerative Morphing provides an on-the-fly morphing renderer trained on two different scenes without any correspondences. The main idea is to extend image morphing from single-view to multiview and generate view-consistent multiview morphs. We introduce an efficient learning strategy in Sec. 4 to derive volume representations from multiview images. We use optimal transport in 2–Wasserstein space to merge the volumes of the two scenes. The algorithm for computing the morphs is detailed in Sec. 5.

## 4 Volume Renderer

We lift the image morphing problem from single-view to multiview by learning volumetric representations for the source and target scenes. The representation contains shape and appearance, and is learned from multiview images and their camera poses. Below, we briefly introduce how to reconstruct such a scene representation from the calibrated input images and our design choices.

To obtain volumes as in Eq. (1), we adopt differentiable volume rendering to optimize the opacity and color volumes for each scene. Given the image poses $\zeta^{\mathcal{S}}$ and $\zeta^{\mathcal{T}}$, we assume a pinhole camera and generate rays emitted from the camera center, based on each pixel's position. During rendering, points are sampled along a ray and queried with the scene representation to produce a series of colors and volume densities. The densities are converted into alpha values via $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$ for the follow-up alpha compositing to accumulate the point queries into a single ray color:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i \ , \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \, , \tag{3}$$

where $\mathbf{r}$ is the camera ray on which the $N$ discrete points are sampled, $T_i$ is the accumulated transmittance from ray emission to the current sample $i$, and $\delta_i$ is the distance between adjacent samples. The scene representation is optimized by minimizing the photometric mean squared error

$$\mathcal{L} = \frac{1}{K} \sum_{m=1}^{K} \left\| \hat{C}(\mathbf{r}_m) - C(\mathbf{r}_m) \right\|_2^2 \ , \tag{4}$$

where $K$ is the mini-batch size, $\hat{C}$ is the rendered color, and $C$ is the observed pixel color.

There are two common volumetric representations: *i*) voxel grids, which explicitly parameterize the 3D scene as grid values, and *ii*) multilayer perceptrons (MLP), which implicitly learn the mapping via MLP weights. We opt to use the explicit voxel grid to model the scene for faster convergence and for the convenience of latter usage in morphing. During training, we learn volumes of opacity $\mathcal{V}_\alpha(\mathbf{x})$ and color $\mathcal{V}_\mathbf{c}(\mathbf{x})$ for each scene, while each sample on the rays is trilinearly interpolated with neighboring voxels. We note, however, that the trained implicit representations [21, 61] can easily be turned into volumetric representations and used with our morphing algorithm.

## 5 Wasserstein Morphing Flow

With learned volumetric representations $\mathcal{V}^{\mathcal{S}}$ and $\mathcal{V}^{\mathcal{T}}$, we develop a differentiable morphing algorithm that welds two volumes into a morphed volume for rendering. Since the source and target scenes are not constrained to be in one category and may be very dissimilar, optimal transport is used to deform the source scene into the target. Specifically, we use Sinkhorn divergence (SD), a regularized optimal transport objective that minimizes 2–Wassertein distance between two point sets. The source and target shapes $\mathcal{S}$ and $\mathcal{T}$ are created as weighted point sets collected from the volumes $\mathcal{V}^{\mathcal{S}}$ and $\mathcal{V}^{\mathcal{T}}$. Note that our method assumes the two weighted point sets to be positive discrete measures such that they can be compared in Wasserstein metrics. Therefore, the weights of the point sets have been normalized to make them discrete probability distributions, *i.e.*, the weights $\{\omega_i^{\mathcal{S}}\}_{i=1}^{N^{\mathcal{S}}}$ and $\{\omega_j^{\mathcal{T}}\}_{j=1}^{N^{\mathcal{T}}}$ of the source and target shapes satisfy $\sum_i^{N^{\mathcal{S}}} \omega_i^{\mathcal{S}} = 1$ and $\sum_j^{N^{\mathcal{T}}} \omega_j^{\mathcal{T}} = 1$. More precisely, such a discrete measure can be expressed as a sum of weighted Dirac mass, and we thus have $\mathcal{S} = \sum_{i=1}^{N^{\mathcal{S}}} \omega_i^{\mathcal{S}} \Delta_{\mathbf{x}_i^{\mathcal{S}}}$ and $\mathcal{T} = \sum_{j=1}^{N^{\mathcal{T}}} \omega_j^{\mathcal{T}} \Delta_{\mathbf{x}_j^{\mathcal{T}}}$, where $\Delta$ is the Dirac delta function that can be thought of as an indicator of occupancy at a given point $\mathbf{x}$.

Our aim now is to obtain a 3D morphing renderer, where at the core is a morphed volumetric representation $\mathcal{V}^{\mathcal{M}_t}$. We view the morphing process as solving an optimal transport problem for moving mass from a source distribution to a target distribution. To generate smooth transitions, we design a flow-based morphing scheme, which comprises the rigid transformation (RT) flow and optimal transport (OT) flow. The RT flow pushes the source scene toward the target scene by applying rotation and translation. As the rigid transformation is global, it can preserve the original appearance. On the other hand, the OT flow provides smooth local deformations but may change the topology of the shape. In what follows, we first describe how to compute the RT flow for globally registering the two shapes, and then we detail the algorithm of OT flow, as well as the complete scheme of dual-flow based morphing.

### 5.1 Rigid transformation flow

With the two shapes $\mathcal{S}$ and $\mathcal{T}$ expressed as two discrete measures that are derived from the source and target volumes, we estimate the rigid transformation $\Psi \in \mathrm{SE}(3)$ by minimizing the Sinkhorn divergence SD [18] between the transformed source measure and the target measure:

$$\hat{\Psi} = \arg\min_{\Psi} \ \mathrm{SD}\left(\Psi(\mathcal{S}), \mathcal{T}\right) \ , \tag{5}$$

where the transformed measure $\Psi(\mathcal{S})$ is defined in the form of weighted sum of Dirac mass by

$$\Psi(\mathcal{S}) = \sum_{i=1}^{N^{\mathcal{S}}} \omega_i^{\mathcal{S}} \, \Delta_{\Psi(\mathbf{x}_i^{\mathcal{S}})} \, . \tag{6}$$

The rigid transformation $\Psi$ comprises rotation $\mathbf{R}$ and translation $\mathbf{z}$. The initial states $\mathbf{R}^{(0)} = \mathbf{I}_3$ and $\mathbf{z}^{(0)} = \mathbf{0}$ are updated by the gradients $\nabla_{\mathbf{R}} \, \mathrm{SD}(\Psi(\mathcal{S}), \mathcal{T})$ and $\nabla_{\mathbf{z}} \, \mathrm{SD}(\Psi(\mathcal{S}), \mathcal{T})$. Note that directly applying the gradient update to $\mathbf{R}$ may lead to an unconstrained projection matrix, and therefore we replace the gradient-updated matrix with identity singular values via singular value decomposition

(SVD). Specifically, we compute

$$(\mathbf{R} - \nabla_{\mathbf{R}} \, \mathrm{SD}(\Psi(\mathcal{S}), \mathcal{T})) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\intercal} \, , \tag{7}$$

and use $\mathbf{U}\mathbf{V}^{\intercal}$ as a surrogate for the new rotation matrix. Finally, we can solve Eq. (5) for the estimated transformation $\hat{\Psi}$, and the transformation flow $\mathbf{f}$ parameterized by the time step $t \in [0, 1]$ is then given by

$$\mathbf{f}(t) = t \cdot (\hat{\Psi}(\mathcal{S}) - \mathcal{S}) \, , \tag{8}$$

which is used in Eq. (2) to provide the progression on the source shape $\mathcal{S}$. For brevity, the definition of Sinkhorn divergence SD and the derivation of the gradients $\nabla_{\mathbf{R}} \, \mathrm{SD}$ and $\nabla_{\mathbf{z}} \, \mathrm{SD}$ are omitted here. More details can be found in the supplementary material.

### 5.2 Optimal transport flow

The rigid transformation makes the two measures $\hat{\Psi}(\mathcal{S})$ and $\mathcal{T}$ distribute in similar loci in $\mathbb{R}^3$. We can further find a smooth deformation between them using optimal transport. In our method, the interpolation is achieved by adding the gradient that minimizes the Sinkhorn divergence between the transformed shape and the target. Given a time step $t \in [0, 1]$ as a blending weight, the optimal transport flow can be written as

$$\mathbf{g}(t) = -t \cdot \nabla_{\mathbf{x}} \, \mathrm{SD}(\hat{\Psi}(\mathcal{S}), \mathcal{T}) \, , \tag{9}$$

which facilitates on-the-fly rendering with a varying time step $t$. Instead of applying the two flows $\mathbf{f}$ and $\mathbf{g}$ sequentially, we advance the two flows simultaneously on $t \in [0, 1]$ as shown in Eq. (2), and get the morphed measure $\mathcal{M}_t$ as the morphed shape. The disentanglement of the two flows in Wasserstein metrics enables the morphing to evolve as rigidly as possible even under inevitable topology changes.

We voxelize the morphed measure $\mathcal{M}_t$ into $\mathcal{V}_{\alpha}^{\mathcal{M}_t}$ by collecting the points with histograms. Each point allocates its weight to the bin according to its position. The histogram is then transformed to a volume. The discretization has to be implemented with care to prevent aliasing. Here we first spread each point in $\mathcal{M}_t$ to its eight neighbors on the grid. The weight also splits into eight according to the distances between points. The histogram gathers weights of all points and generates $\mathcal{V}_{\alpha}^{\mathcal{M}_t}$. $\hat{\mathcal{M}}_{t=1}$ is used to query the color volume $\mathcal{V}_{\mathbf{c}}^{\mathcal{T}}$ for the corresponding target colors. The color of each morphed point is blended between the source color and target color. The morphed color volume $\mathcal{V}_{\mathbf{c}}^{\mathcal{M}_t}$ is generated using a histogram method similar to $\mathcal{V}_{\alpha}^{\mathcal{M}_t}$.

## 6 Results

We evaluate our method on real and synthetic datasets, including **Synthetic–NeRF** [42], **Synthetic–NSVF** [37], **Tanks&Temples** [32] and **BlendedMVS** [60]. Each datasets contains scenes with surrounded imaging and their camera poses.

### 6.1 Multiview regenerative morphing

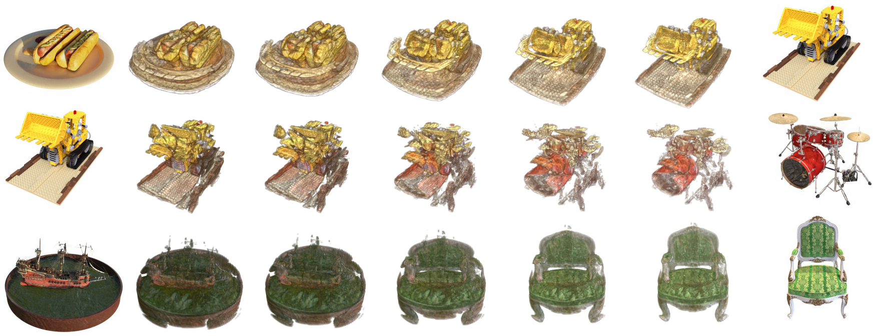

**Fig. 4.** Morphing of scenes in Synthetic–NeRF. Each row shows a smooth transition from the source (left) to the target (right).

Each scene may contain a number of objects, where the morphing between scenes needs to divide or merge objects smoothly. Fig. 1 shows morphed images from materials with different colors and reflections into a single microphone. The top and bottom rows show the different views of the morph frozen at $t = 0.3$ and $t = 0.7$. The rendering results are view-consistent, *i.e.*, at the frozen moment, the intermediate morph can be viewed as an actual coherent scene and exhibits no conflicts across different views.

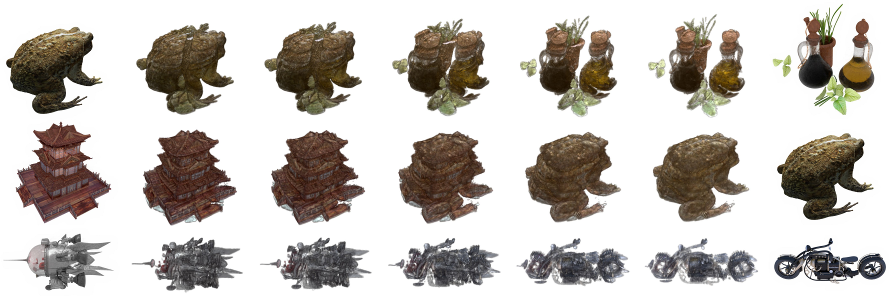

**Fig. 5.** Morphing of scenes in Synthetic–NSVF. Each row shows a smooth transition from the source (left) to the target (right).

Fig. 4 shows morphing with scenes in Synthetic-NeRF. We demonstrate smooth transitions between three different sets of source and target scenes. Especially, in the middle row a lego truck morphs into a drum set, with very complex detail. In Fig. 5, we show morphing results in Synthetic-NSVF. Due to the limitation of space, we provide more results of multiview rendering in videos in the supplementary material.

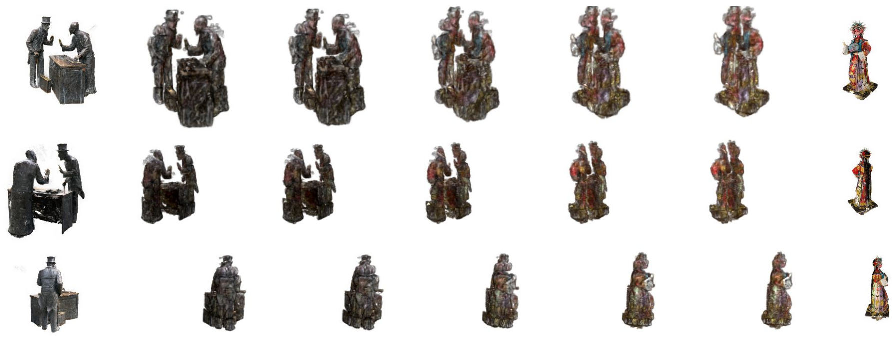

**Fig. 6.** Morphing of scenes in BlendedMVS. All the rows exhibit the same transition, while each row shows renderings under some view.

Fig. 6 demonstrates the morphing between 'Statue' and 'Character'. Both scenes are from BlendedMVS. The three rows represent the same transition rendered in different viewpoints. Two people in the Statue gradually get close to each other and merge into a single person. Also the scene becomes colorful, from metallic texture into custom and makeup. Since the training images have abundant specular lighting, resulting in noises and heavy shadows, we therefore use thresholding and color distribution manipulation on the morphed volumes to compensate the flaw.

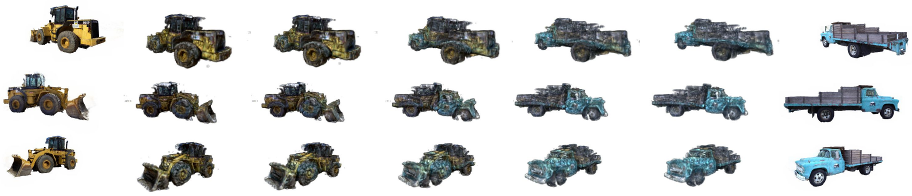

**Fig. 7.** Morphing of scenes in Tanks&Temples. All the rows exhibit the same transition, but in different views.

Fig. 7 shows transitions between 'Caterpillar' and 'Truck'. Both scenes are from Tanks&Temples. The two shapes are reconstructed with collected images, under varying lighting conditions. The direct result has the floating noise around the shapes. We use simple thresholding to remove the noise in the space of the volume. Each row shows renderings of the transition under some view. The morphs are view-consistent in each column. Each column relates to some blending

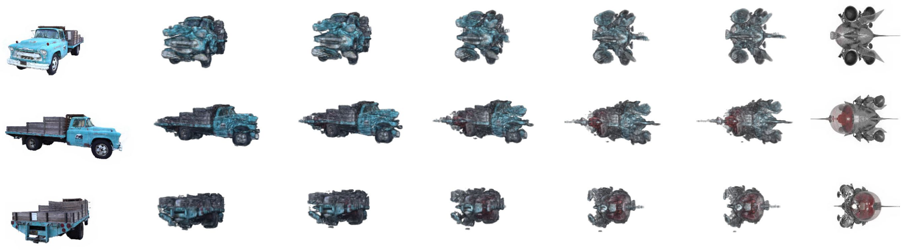

**Fig. 8.** Morphing between real and synthetic scenes. We demonstrate view-consistent morphing by rendering in three different views.

weight. In addition to morphing real scenes, we demonstrate morphing between real and synthetic scenes. Fig. 8 shows the morphs from a real truck to synthetic 'Spaceship'. Likewise, Fig. 9 interpolates between 'Caterpillar' and 'Character' from BlendedMVS. The real to synthetic scene morphing is achieved under our normal setting.

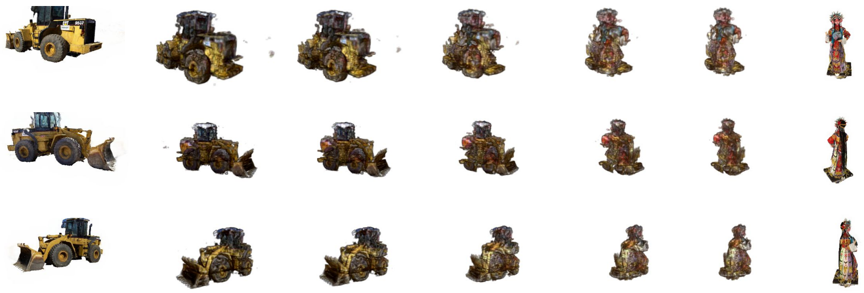

**Fig. 9.** Morphing between real and synthetic scenes. The morphs are rendered with three viewpoints, one in each row.

**Limitations on real scenes.** Our dual-flow multiview morphing is a model-free method and therefore not restricted to specific data domains. Fig. 7 shows the morphing between two real scenes, where our method generates reasonable transitions between the two very dissimilar scenes. However, due to the lighting changes across views, the rendered morphs contain more noise in comparison with synthetic data. The quality of multiview morphing relies on the reconstructed volumetric representations of the original scenes. Those reconstruction artifacts tend to remain during the entire morphing process. Similar artifacts are observed in Fig. 6, where the reflection of the surface affects the reconstruction.

### 6.2 Comparisons with other approaches

We validate that our morph generation is geometry-aware in contrast to other correspondence-free morphing algorithms. We compare our method with a 3D-based method, Debiased Sinkhorn [26], and a 2D-based method, Deep Image Analogy [16]. To compare with the 2D-based method, we use the optimized volume of each scene to generate pose-aligned images as their input. NeRF's eight scenes are used for evaluation. We randomly sample camera poses from the upper hemisphere for different scenes and render the morphs with varying transition weights. For each transition, we use COLMAP to solve *structure from motion*. As a result, 83.3% of the morphing images generated by our method are successfully registered by COLMAP, which means our method can mostly render 3D consistent morphs. On the other hand, Debiased Sinkhorn [26] generates blur images, resulting in poor reconstruction: Only 16.7% of its morphing images can be successfully registered by COLMAP. Deep Image Analogy [16] generates visually pleasing morphs, but it is not robust to view changes, as mentioned in [47]: not surprisingly, no consistent 3D structure can be reconstructed.

### 6.3 Ablation study

We evaluate different aspects of our method, especially the effect of the RT flow and the OT flow. We compare the rendering of direct optimal-transport morphing with our rigid-transformation-enabled OT morphing. We also demonstrate the effect when only one of the RT flow or the OT flow is applied.

**Rigid transformation flow:** As mentioned in [2,14,46], achieving as-rigid-as-possible transformation is an appealing property for morphing. The property preserves the original shape during transition and thus helps to produce plausible intermediate results. Unlike previous methods that generate meshes for deformation, here we formulate the rigid transformation as a flow in Wasserstein space. Fig. 10 shows comparisons of transitions with or without the rigid transformation flow. The first row shows renderings using our full model with both the RT and OT flows. The second row shows renderings without the RT flow; only the OT flow is used. It can be seen that morphing without the RT flow directly moves each point toward the target and results in shattered rendering. Such an effect is unsatisfactory as the edge of the plate falls into pieces. In contrast, our method gradually transforms the plate so that the hotdog and the ficus have their poses aligned. During transformation, the OT flow simultaneously performs deformation in local regions so the texture can better resemble the target. We also compare to the baseline, where no flows but simple blending is applied, as shown in the third row. Morphing with cross-dissolve leads to ghost effects.

**Optimal transport flow:** Our flow composition method generates smooth morphing by jointly performing rigid transformation and OT deformation. Here we examine the effect when only one of the two flows is used. This can be manipulated by the blending weight $t$ in each flow. As previously shown in Fig. 3,

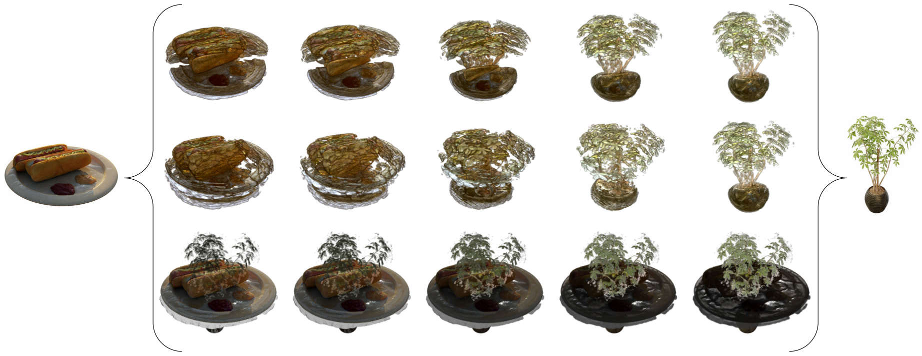

**Fig. 10.** Comparisons between three morphing method. First row shows our dual-flow morphing. Second row uses only OT flow. Third row use no flow, but simple blending.

we use only the RT flow in the first half of morphing, and the OT flow in the second half. The source scene is on the left, and the subsequent three images are only affected by the RT flow. We can see that the flow aligns the poses of the source and target shapes. When the two scenes are fully aligned, the OT flow deforms the hotdog into the ficus, as shown in the fifth and sixth images.

## 7 Conclusion

This paper introduces Multiview Regenerative Morphing—a new method that integrates volume rendering and optimal transport to address a new task of multiview image morphing. Our method can produce interesting morphing effects that have not yet been demonstrated by previous image-based morphing methods. From the multiview images of two category-agnostic scenes without predefined correspondences, our method learns volumetric representations to render free-form morphs that can be visualized from arbitrary perspectives at any transition moment. We decouple the morphing process into two flows in Wasserstein metrics: one governs the rigid transformation and the other models the correspondences and deformations. The two flows estimated via optimization then jointly provide as-rigid-as-possible transformation under required topological and morphological changes between the two shapes. Our method is fast in training; it takes less than half an hour to learn the morphing renderer from both scenes, which otherwise might need 30x longer time if learned by typical neural rendering methods. The learned morphing renderer can readily generate on-the-fly multiview morphs showcasing new visual effects.

**Acknowledgements:** This work was supported in part by the MOST grants 110-2634-F-007-027 and 111-2221-E-001-011-MY2 of Taiwan. We are grateful to National Center for High-performance Computing for providing computational resources and facilities.

## References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 4431–4440. IEEE (2019) 3
2. Alexa, M., Cohen-Or, D., Levin, D.: As-rigid-as-possible shape interpolation. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 157–164 (2000) 3, 13
3. 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: ICCV (2021) 4
4. Benamou, J.D., Carlier, G., Cuturi, M., Nenna, L., Peyré, G.: Iterative bregman projections for regularized transportation problems. SIAM Journal on Scientific Computing **37**(2), A1111–A1138 (2015) 3
5. Bonneel, N., Peyré, G., Cuturi, M.: Wasserstein barycentric coordinates: histogram regression using optimal transport. ACM Trans. Graph. **35**(4), 71–1 (2016) 3
6. Bonneel, N., Van De Panne, M., Paris, S., Heidrich, W.: Displacement interpolation using lagrangian mass transport. In: Proceedings of the 2011 SIGGRAPH Asia Conference. pp. 1–12 (2011) 3
7. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: ICCV (2021) 4
8. Cheng, S., Bronstein, M.M., Zhou, Y., Kotsia, I., Pantic, M., Zafeiriou, S.: Meshgan: Non-linear 3d morphable models of faces. arxiv CS.CV 1903.10384 (2019) 4
9. Cosmo, L., Norelli, A., Halimi, O., Kimmel, R., Rodolà, E.: LIMP: learning latent shape representations with metric preservation priors. In: Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12348, pp. 19–35. Springer (2020) 4
10. Darabi, S., Shechtman, E., Barnes, C., Goldman, D.B., Sen, P.: Image melding: Combining inconsistent images using patch-based synthesis. ACM Transactions on graphics (TOG) **31**(4), 1–10 (2012) 3
11. Deng, K., Liu, A., Zhu, J., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. arxiv CS.CV 2107.02791 (2021) 4
12. Eisenberger, M., Cremers, D.: Hamiltonian dynamics for real-world shape interpolation. In: Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 12349, pp. 179–196. Springer (2020) 4
13. Eisenberger, M., Lähner, Z., Cremers, D.: Divergence-free shape correspondence by deformation. Comput. Graph. Forum **38**(5), 1–12 (2019) 4
14. Eisenberger, M., Novotny, D., Kerchenbaum, G., Labatut, P., Neverova, N., Cremers, D., Vedaldi, A.: Neuromorph: Unsupervised shape interpolation and correspondence in one go. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7473–7483 (2021) 4, 13
15. *et al.*, B.: Poxels: Probabilistic voxelized volume reconstruction. In: ICCV (1999) 3
16. *et al.*, L.: Visual attribute transfer through deep image analogy. arXiv:1705.01088 (2017) 13
17. *et al.*, S.: Stereo matching with transparency and matting. IJCV (1999) 3

18. Feydy, J., Séjourné, T., Vialard, F.X., Amari, S.i., Trouvé, A., Peyré, G.: Interpolating between optimal transport and mmd using sinkhorn divergences. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 2681–2690. PMLR (2019) 4, 6, 8
19. Fish, N., Zhang, R., Perry, L., Cohen-Or, D., Shechtman, E., Barnes, C.: Image morphing with perceptual constraints and STN alignment. Comput. Graph. Forum **39**(6), 303–313 (2020) 3
20. Gao, C., Saraf, A., Kopf, J., Huang, J.: Dynamic view synthesis from dynamic monocular video. In: ICCV (2021) 4
21. Garbin, S.J., Kowalski, M., Johnson, M., Shotton, J., Valentin, J.P.C.: Fastnerf: High-fidelity neural rendering at 200fps. arxiv CS.CV 2103.10380 (2021) 4, 7
22. Genevay, A., Peyré, G., Cuturi, M.: Learning generative models with sinkhorn divergences. In: International Conference on Artificial Intelligence and Statistics. pp. 1608–1617. PMLR (2018) 6
23. Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.E.: Baking neural radiance fields for real-time view synthesis. In: ICCV (2021) 4
24. Heeren, B., Rumpf, M., Schröder, P., Wardetzky, M., Wirth, B.: Splines in the space of shells. Comput. Graph. Forum **35**(5), 111–120 (2016) 4
25. Heeren, B., Rumpf, M., Wardetzky, M., Wirth, B.: Time-discrete geodesics in the space of shells. Comput. Graph. Forum **31**(5), 1755–1764 (2012) 4
26. Janati, H., Cuturi, M., Gramfort, A.: Debiased sinkhorn barycenters. In: International Conference on Machine Learning. pp. 4692–4701. PMLR (2020) 4, 13
27. Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., Park, J.: Self-calibrating neural radiance fields. In: ICCV (2021) 4
28. Jiang, C.M., Marcus, P.: Hierarchical detail enhancing mesh-based shape generation with 3d generative adversarial network. arxiv CS.CV 1709.07581 (2017) 4
29. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 4401–4410. Computer Vision Foundation / IEEE (2019) 2
30. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 8107–8116. Computer Vision Foundation / IEEE (2020) 2
31. Kilian, M., Mitra, N.J., Pottmann, H.: Geometric modeling in shape space. ACM Trans. Graph. **26**(3), 64 (2007) 4
32. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017) 9
33. Lerios, A., Garfinkle, C.D., Levoy, M.: Feature-based volume metamorphosis. In: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. pp. 449–456 (1995) 3
34. Li, C., Zaheer, M., Zhang, Y., Póczos, B., Salakhutdinov, R.: Point cloud GAN. In: Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019. OpenReview.net (2019) 4
35. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: CVPR (2021) 4
36. Lin, C., Ma, W., Torralba, A., Lucey, S.: BARF: bundle-adjusting neural radiance fields. In: ICCV (2021) 4
37. Liu, L., Gu, J., Lin, K.Z., Chua, T., Theobalt, C.: Neural sparse voxel fields. In: NeurIPS (2020) 9

38. Liu, Y., Peng, S., Liu, L., Wang, Q., Wang, P., Theobalt, C., Zhou, X., Wang, W.: Neural rays for occlusion-aware image-based rendering. arxiv CS.CV 2107.13421 (2021) 4
39. 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: CVPR (2021) 4
40. Meng, Q., Chen, A., Luo, H., Wu, M., Su, H., Xu, L., He, X., Yu, J.: Gnerf: Gan-based neural radiance field without posed camera. In: ICCV (2021) 4
41. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 3
42. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: The European Conference on Computer Vision (ECCV) (2020) 9
43. Oechsle, M., Peng, S., Geiger, A.: UNISURF: unifying neural implicit surfaces and radiance fields for multi-view reconstruction. arxiv CS.CV 2104.10078 (2021) 4
44. Pan, X., Zhan, X., Dai, B., Lin, D., Loy, C.C., Luo, P.: Exploiting deep generative prior for versatile image restoration and manipulation. In: Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12347, pp. 262–277. Springer (2020) 3
45. Park, J.J., Florence, P., Straub, J., Newcombe, R.A., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 165–174. Computer Vision Foundation / IEEE (2019) 4
46. Schaefer, S., McPhail, T., Warren, J.: Image deformation using moving least squares. In: ACM SIGGRAPH 2006 Papers, pp. 533–540 (2006) 13
47. Seitz, S.M., Dyer, C.R.: View morphing. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. pp. 21–30 (1996) 2, 3, 13
48. Shechtman, E., Rav-Acha, A., Irani, M., Seitz, S.: Regenerative morphing. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 615–622. IEEE (2010) 2, 3
49. Shu, D.W., Park, S.W., Kwon, J.: 3d point cloud generative adversarial network based on tree structured graph convolutions. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 3858–3867. IEEE (2019) 4
50. Simon, D., Aberdam, A.: Barycenters of natural images - constrained wasserstein barycenters for image morphing. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 7907–7916. Computer Vision Foundation / IEEE (2020) 3
51. Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., Guibas, L.: Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. ACM Transactions on Graphics (TOG) **34**(4), 1–11 (2015) 3, 4
52. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. arXiv preprint arXiv:2111.11215 (2021) 4
53. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.A.: Ibrnet: Learning multi-view image-based rendering. In: CVPR (2021) 4

54. Wirth, B., Bar, L., Rumpf, M., Sapiro, G.: A continuum mechanical approach to geodesics in shape space. Int. J. Comput. Vis. **93**(3), 293–318 (2011) 4
55. Wizadwongsa, S., Phongthawee, P., Yenphraphai, J., Suwajanakorn, S.: Nex: Real-time view synthesis with neural basis expansion. In: CVPR (2021) 4
56. Wolberg, G.: Image morphing: a survey. The visual computer **14**(8), 360–372 (1998) 3
57. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 82–90 (2016) 4
58. Wu, Z., Nitzan, Y., Shechtman, E., Lischinski, D.: Stylealign: Analysis and applications of aligned stylegan models. arxiv CS.CV 2110.11323 (2021) 3
59. Xian, W., Huang, J., Kopf, J., Kim, C.: Space-time neural irradiance fields for free-viewpoint video. In: CVPR (2021) 4
60. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1790–1799 (2020) 9
61. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: ICCV (2021) 4, 7
62. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelNeRF: Neural radiance fields from one or few images. https://arxiv.org/abs/2012.02190 (2020) 4
63. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arxiv CS.CV 2010.07492 (2020) 4