










PREF: Predictability Regularized Neural Motion Fields

Liangchen Song^{1,2}, Xuan Gong^{1,2}, Benjamin Planche²,
Meng Zheng², David Doermann¹, Junsong Yuan¹, Terrence Chen²,
and Ziyang Wu²

¹ University at Buffalo, Buffalo, NY, USA

² United Imaging Intelligence, Cambridge, MA, USA

{liangchen.song, xuan.gong, benjamin.planche,
meng.zheng, ziyang.wu}@united-imaging.com

Abstract. Knowing the 3D motions in a dynamic scene is essential to many vision applications. Recent progress is mainly focused on estimating the activity of some specific elements like humans. In this paper, we leverage a neural motion field for estimating the motion of all points in a multiview setting. Modeling the motion from a dynamic scene with multiview data is challenging due to the ambiguities in points of similar color and points with time-varying color. We propose to regularize the estimated motion to be predictable. If the motion from previous frames is known, then the motion in the near future should be predictable. Therefore, we introduce a predictability regularization by first conditioning the estimated motion on latent embeddings, then by adopting a predictor network to enforce predictability on the embeddings. The proposed framework PREF (**P**redictability **R**Egularized **F**ields) achieves on par or better results than state-of-the-art neural motion field-based dynamic scene representation methods while requiring no prior knowledge of the scene.

Keywords: Neural fields · Motion estimation · Motion prediction

1 Introduction

Estimating motion in dynamic scenes is a fundamental and long-standing problem in computer vision [16]. Most of the existing 3D motion estimation works are concerned with specific objects like humans [42]. Still, knowing the 3D motion of all objects in a dynamic scene can be of great benefit to a number of vision applications like robot path planning [8]. Tracking all points in the space with only multiview data is obviously challenging, however, neural fields is a hot topic that has emerged recently [59], bringing hope to breakthroughs for this problem.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-20047-2_38.

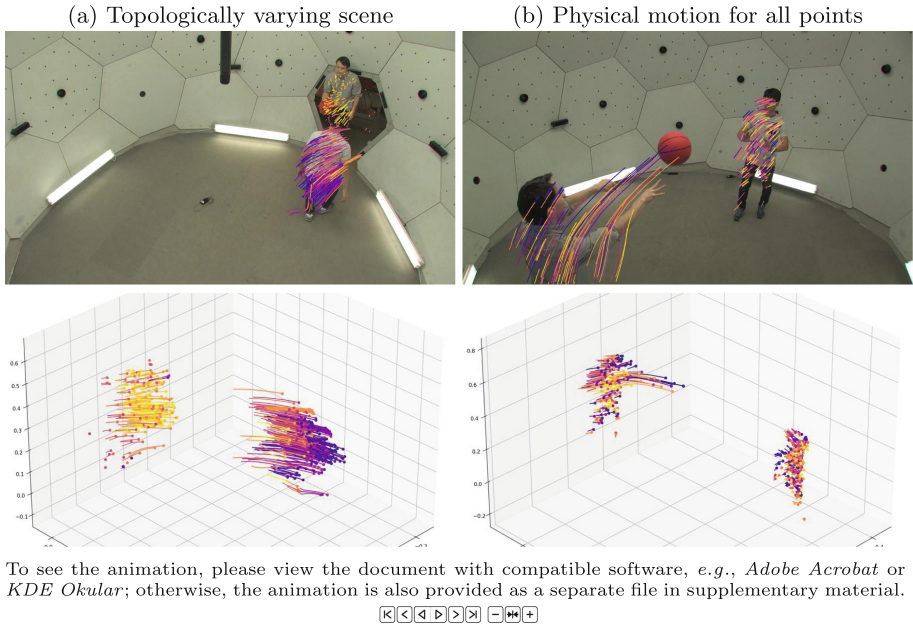


Fig. 1. Our method can handle topologically varying scenes and estimate physical motion for all points in the space. *Topologically varying* means that the topology of the scene can change, such as a new person entering the scene in (a). All points in the space are tracked, such as the ball in (b). Only the sequence of images to be analyzed is used and no prior knowledge is required in our framework. (See supplementary material for an animated version of the figure.)

Neural fields, also known as coordinate-based neural networks, have demonstrated great potential in dynamic 3D scene reconstruction from multiview data [51, 59]. Coordinate-based representations not only naturally support fine-grained modeling of the motion for points in space, but also require no prior knowledge about the geometry and track all points in space. In this paper, we address the problem of estimating 3D motion from multiview image sequences, for general scenes and for all points in the space (Fig. 1).

Despite recent progress on neural fields-based dynamic scene representation (e.g., [9–11, 20, 21, 25, 36, 37, 41, 52, 56, 57, 63]), estimating 3D motion from multiview data remains challenging for the following reasons. First, motion ambiguity exists among points with the same color, so one cannot confidently track interchangeable points on non-rigid surfaces from visual observations alone (*c.f.* possibility of position swapping). Second, the color of any point may change over time. For example, spatially or temporally varying lighting conditions can blur the notion of a point’s identity over time.

In this paper, we propose to *regularize the estimated motion to be predictable* to address the aforementioned ambiguity issues. The key insight behind motion predictability is that underlying motion patterns exist in a dynamic real-world

scene. Chaotic motions (*e.g.*, position swapping for similarly-colored points) are not predictable and should be penalized. In our work, the motion in a scene is “*implicitly*” regularized by enforcing predictability, which is intrinsically different from explicitly designed regularizing terms, such as elastic regularization [36] and as-rigid-as-possible regularization [52].

State-of-the-art solutions use combinations of space-time radiance neural fields and neural motion fields to model dynamic scenes, optimizing these fields jointly over a set of visual observations in a self-supervised manner by comparing predicted images to actual observations. But vision-based supervision alone typically results in noisy and poorly disentangled motion fields, *c.f.* aforementioned ambiguities. Therefore, some recent works use data-driven priors like depth [57] and 2D optical flow [21] as a regularization. In contrast, we propose to improve motion field optimization through predictability-based regularization. Instead of learning a motion field M that maps each 3D position \mathbf{p} and timestep t to a deformation vector $\Delta_{t \rightarrow t+\delta t}\mathbf{p}$, we condition the motion field on a predictable embedding of the motion for queried time (noted $\omega_{t \rightarrow t+\delta t}$), *i.e.*, $\Delta_{t \rightarrow t+\delta t}\mathbf{p} = M(\mathbf{p}, \omega_{t \rightarrow t+\delta t})$. These motion embeddings are either directly optimized jointly with the space-time field over observations, or are inferred by a predictor function P that takes a set of past embeddings and infers the next motion embedding. During scene optimization, we enforce each motion embedding regressed from the observations to be predictable by our model P . Therefore we promote the encoding of underlying motion patterns and penalize chaotic and unlikely-realistic deformations. In summary, our contributions are as follows:

- We propose to leverage predictability as a prior w.r.t. the motion in a dynamic scene. Predictability regularization implicitly penalizes chaotic motion estimation and can help solve the ambiguity of motion.
- We condition point motions on embedding vectors and design a predictor on the embedding space to enforce motion predictability.
- We demonstrate the benefits of the resulting additional supervision (predictability regularization) on motion learning through a variety of qualitative and quantitative evaluations.
- We provide insights into how the proposed framework can be leveraged for motion prediction as a by-product.

2 Related Work

Neural Fields. A neural field is a field that is parameterized fully or in part by a neural network [6, 59]. Neural fields are widely used for implicitly encoding the geometry of a scene, such as occupancy [29] and distance function [7, 35]. Our method is built on the milestone work NeRF [30], in which the radiance and density are encoded in neural fields. NeRF led to a series of breakthroughs in the fields of 3D scene understanding and rendering, such as relighting [2, 3, 46], human face and body capture [14, 24, 34, 39, 40, 49], and city-scale reconstruction [43, 50, 53, 58]. A recent method also named PREF [15] is developed for compact neural signal modeling.

Motion Estimation and 4D Reconstruction. Large-scale learning-based motion estimation from multiview data achieved impressive performance [22, 42], but most methods are constrained to tracking some specific objects such as humans [42]. In this paper, we are concerned with estimating the motion of all points without access to any annotations, which is related to the 4D reconstruction problem where motion is usually estimated. Some methods have been developed with known geometry information such as depth or point cloud. Dynamic-Fusion [32], Schmidt *et allet@tokeneonedot*[44], Bozic *et allet@tokeneonedot*[5], and Yoon *et allet@tokeneonedot*[61] estimate motion from videos with depth. OFlow [33] and ShapeFlow [17] infer a deformation flow field with the knowledge of occupancy. More recently, motivated by the success of NeRF, a number of methods have been designed to reconstruct 4D scenes as well as motion directly from multiview data, which can be acquired from a multi-camera system or a single moving camera. D-NeRF [41], Nerfies [36] and NR-NeRF [52] set a canonical frame and align dynamic points to it. DCT-NeRF [56] proposes to track the trajectory of a point along all sequences. NSFF [21], VideoNeRF [57], and NeRFlow [9] propose to represent the dynamic scene with a 4D space-time field, thus able to handle topologically varying scenes. The 4D fields are under-determined, and precomputed data-driven priors are usually needed to achieve good performance. HyperNeRF [37] proposes to align frames towards a hyperspace for topologically varying scenes and achieves state-of-the-art performance without the need of data-driven priors. These methods are able to render visually appealing images for novel views and time, yet their performance on 3D motion estimation has room for improvements.

Scene Flow Estimation. 3D motion field is also known as dense scene flow [28, 31, 62]. Vedula *et allet@tokeneonedot*[54] introduced the concept and demonstrated a framework for acquiring dense, non-rigid scene flow from optical flow. Basha *et allet@tokeneonedot*[1] proposed a 3D point cloud parameterization of the 3D structure and scene flow with calibrated multi-view videos. Vogel *et allet@tokeneonedot*[55] suggested to represent the dynamic 3D scene by a collection of planar, rigidly moving, local segments. More recently, Yang *et allet@tokeneonedot*[60] proposed a framework adopting 3D rigid transformations for analyzing background segmentation and rigidly moving objects.

Predictability. The study of the predictability of time series data dates back to [4, 38], in which predictability is interpreted as the ability to be decomposed into lower-dimensional components. The idea of extracting principal components as predictability is adopted for blind source separation in [48]. Differential entropy is used for measuring predictability in [12]. Our method shares a similar motivation as the above methods in terms of discovering low-rank structures, while predictability in our method is not explicitly defined but implicitly introduced through a predictor network.

3 Preliminaries

Our method is built upon the NeRF framework [30] and is inspired by recent progresses w.r.t.let@tokeneonedotdynamic scenes [21, 57]. For each 3D point $\mathbf{p} = (x, y, z)$ in the considered space, we represent its volume density by $\sigma(\mathbf{p})$, and its color from a viewing direction \mathbf{d} by $\mathbf{c}(\mathbf{p}, \mathbf{d})$. In NeRF, these two attributes are defined as the output of a continuous function F modeled by a neural network, *i.e.*, $(\mathbf{c}, \sigma) = F(\mathbf{p}, \mathbf{d})$. This neural field can be queried to render images of the represented scene through volume rendering. For each camera ray \mathbf{r} defined by its optical origin \mathbf{o} and direction \mathbf{d} intersecting a pixel, we compute the color $\mathbf{C}(\mathbf{r})$ of said pixel by sampling points along the ray, *i.e.*, sampling $\mathbf{p}_i = \mathbf{o} + i\mathbf{d}$; then querying and accumulating their attributes according to F . Overall, the expected color $\mathbf{C}(\mathbf{r})$ of the ray \mathbf{r} is:

$$\mathbf{C}(\mathbf{r}) = \int_{i_n}^{i_f} e^{-\int_{i_n}^i \sigma(\mathbf{p}_j) dj} \sigma(\mathbf{p}_i) \mathbf{c}(\mathbf{p}_i, \mathbf{d}) di, \quad (1)$$

where i_n, i_f are near and far bounds. The integration in Eq. 1 is numerically approximated by summing up a set of points on the ray.

For dynamic scenes, existing solutions can be roughly categorized into two groups. Either methods model the motion and radiance with two distinct fields [36, 41], or they are regularizing the motion from a space-time field [9, 21, 57]. In the former solutions, the color of a point \mathbf{p} at time t is represented by $F_k(M(\mathbf{p}, t), \mathbf{d})$, where F_k represents the k th canonical time-invariant space and M is a learned neural motion field defining the motion $\Delta\mathbf{p}$ of any point \mathbf{p} at time t w.r.t.let@tokeneonedotto their position in the canonical space. Our method falls into the latter category, in which each point in the dynamic scene is represented by a space-time field $F(\mathbf{p}, \mathbf{d}, t)$. Unlike canonical space-based methods, for the space-time field we need to specify the frame of F when joint training with a motion field M . We opt for space-time field rather than canonical-space one for two reasons. First, we presume that underlying patterns exist for the motion of a certain time range. So canonical-frame-based motion estimation frameworks are not suitable, since their motions are from the predefined canonical frame to another, whereas we need the motion between a certain range of frames. Second, space-time fields are more generic as they can handle non-existent geometry in the canonical frame (*e.g.*, objects entering the scene mid-sequence). Note that for both categories, the scene fields are optimized jointly leveraging observation-based self-supervision, *i.e.*, computing the image reconstruction loss for each time step t as:

$$\mathcal{L}_{\text{rec}} = \sum_{\mathbf{r}} \|\mathbf{C}_{\text{gt}}^t(\mathbf{r}) - \mathbf{C}^t(\mathbf{r})\|_2^2, \quad (2)$$

with \mathbf{C}_{gt}^t is the observed pixel color and \mathbf{C}^t is the color rendered from F and M .

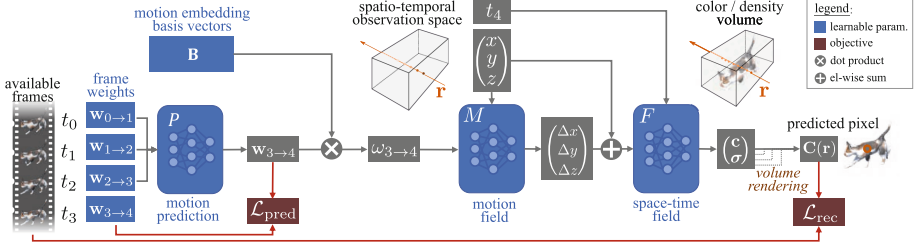


Fig. 2. Overview of the proposed framework. Three networks are trained jointly: the space-time field, the motion field and the predictor. The space-time field returns color and occupancy for each point at a specific time. The motion field predicts the motion of a point based on a motion embedding vector. The predictor generates the future motion embedding based on previously observed embeddings.

4 Method

4.1 Overview

Our framework consists of three components: a neural space-time field F , a motion field M and a motion predictor P . An overview of their interactions is presented in Fig. 2. In our framework and implementations, we do not model the viewing dependency effects with the space-time field, so the space-time field outputs the color and occupancy for each point (x, y, z, t) , whereas the motion field provides the motion of any point between two time steps, according to the space-time field. Let the motion of point $\mathbf{p} = (x, y, z)$ from time t to $t + \delta t$ be $\Delta_{t \rightarrow t + \delta t} \mathbf{p}$, then for \mathbf{p} at time t we have:

$$(\mathbf{c}_t, \boldsymbol{\sigma}_t) = F(\mathbf{p} + \Delta_{t \rightarrow t + \delta t} \mathbf{p}, t + \delta t). \quad (3)$$

The idea is that for a scene observed at time $t + \delta t$, we can obtain the attributes of \mathbf{p} at time t by querying the space-time field with the point location at $t + \delta t$.

In our framework, the motion network is conditioned on an embedding vector $\boldsymbol{\omega}$ (instead of queried timestep) and the motion can be written as $\Delta_{t \rightarrow t + \delta t} \mathbf{p} = M(\mathbf{p}, \boldsymbol{\omega}_{t \rightarrow t + \delta t})$, where $\boldsymbol{\omega}_{t \rightarrow t + \delta t}$ depends on time t and interval δt . Replacing the temporal variable t with a vector $\boldsymbol{\omega}$ as input to M enables predictability via embedding, as further detailed in Sect. 4.2. All networks and the embedding vector w.r.t.let@tokenonedottime t are optimized using the reconstruction loss \mathcal{L}_{rec} (c.f. Eq. 2), with color C^t predicted from $F, M, \boldsymbol{\omega}_{t \rightarrow t + \delta t}$ according to Eqs. 1 and 3.

We define the predictor P as a function taking as input several motion embedding vectors of previous frames and inferring the motion embedding vectors for the future frames accordingly. Mathematically, we have $\boldsymbol{\omega}_{t \rightarrow t + \delta t} = P(\boldsymbol{\omega}_{\text{prev}})$ with $\boldsymbol{\omega}_{\text{prev}} = \{\boldsymbol{\omega}_{t - (i+1)\delta t \rightarrow t - i\delta t}\}_{i=1}^{\tau}$ set of τ previous frames’ embeddings. For example, in Fig. 2, the embedding vector $\boldsymbol{\omega}_{3 \rightarrow 4}$ for motion from t_3 to t_4 is predicted from previous three embedding vectors, that is, $P(\{\boldsymbol{\omega}_{0 \rightarrow 1}, \boldsymbol{\omega}_{1 \rightarrow 2}, \boldsymbol{\omega}_{2 \rightarrow 3}\})$.

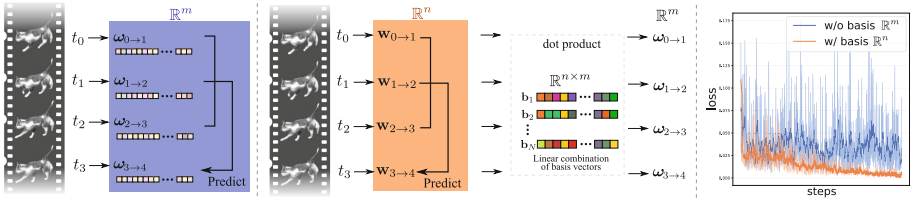


Fig. 3. We use a set of basis vectors for the motion embedding (middle), rather than associating each frame with a motion vector (left). The input and out space of the predictor switches to the linear combination weights by using these shared basis vectors. The comparison of training losses (right) indicates that the predictor converges faster on the space of linear combination weights.

4.2 Neural Motion Fields with Motion Embedding

The motion field is conditioned on an embedding vector, sampled from a latent space depicting motion patterns. Such embedding can be implemented in various ways. The simplest one is to associate each motion of interest with a trainable embedding vector. This technique has been widely used for conditioning neural fields w.r.t.let@tokeneonedotappearance [26] and deformation [36]. However, empirical studies show that associating each motion with motion embedding frequently and significantly slows down the convergence speed of the predictor, as demonstrated in Fig. 3. We presume that the phenomenon is caused by the large and unstructured solution space brought by frame-wise motion embedding. To validate the assumption and improve the convergence speed, we propose to reduce the dimension of the input and output space of the predictor.

Inspired by mixture-of-experts-based prediction networks [13,23,47], we design a set $\mathbf{B} \in \mathbb{R}^{n \times m}$ of n embedding basis vectors, *i.e.*, $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n]^T$ with $\mathbf{b}_i \in \mathbb{R}^m$ basis vector. \mathbf{B} is shared across all frames. Then the motion embedding becomes $\omega_{t \rightarrow t+\delta t} = \mathbf{w}_{t \rightarrow t+\delta t} \cdot \mathbf{B}$, with $\mathbf{w} \in \mathbb{R}^n$ optimizable linear combination weights. Accordingly, we redefine the model P to receive and predict these weight vectors instead of the embedding ones, thus reducing its input space and output space to \mathbb{R}^n , *i.e.*, with the dimensionality of basis vectors not affecting the predictor anymore. In our experiments, we set $n = 5$ and $m = 32$, so the dimension of the predictor’s output space is reduced from 32 to 5. An illustration and comparison of the training losses between the two schemes are presented in Fig. 3.

4.3 Regularizing with Motion Prediction

Our proposed solution makes it possible to complement the usual self-supervision of space-time neural fields (through visual reconstruction only) by a regularization term over motion. However, while predicting motion embeddings is straightforward, *i.e.*, by simply forwarding the embedding vectors of previous frames into P , leveraging P for the regularization of M is not trivial.

In our framework, motion embeddings can be acquired either from reconstruction, *i.e.*, optimizing each embedding along with other components (*e.g.*, both the motion embedding $\boldsymbol{\omega}_{3 \rightarrow 4} = \mathbf{w}_{3 \rightarrow 4} \cdot \mathbf{B}$ and the motion network $M(\mathbf{p}, \boldsymbol{\omega}_{3 \rightarrow 4})$ can be optimized on observed images at $t = 3, 4$); or through the predictor (*e.g.*, $\boldsymbol{\omega}_{3 \rightarrow 4} = P(\{\mathbf{w}_{t-1 \rightarrow t}\}_{t=1}^3) \cdot \mathbf{B}$). We leverage this redundancy for regularization, *i.e.*, proposing a loss to minimize the difference between the self-supervised embeddings and their corresponding predicted versions:

$$\mathcal{L}_{\text{pred}} = \|P(\mathbf{w}_{\text{prev}}) - \arg \min_{\mathbf{w}_{t \rightarrow t+\delta t}} \mathcal{L}_{\text{rec}}\|_2^2, \text{ where } \mathbf{w}_{\text{prev}} = \{\mathbf{w}_{t-(i+1)\delta t \rightarrow t-i\delta t}\}_{i=1}^{\tau}. \quad (4)$$

In the above equation, the first term $P(\cdot)$ represents the motion embedding predicted according to previous τ frames, and the second term $\arg \min_{\mathbf{w}_{t \rightarrow t+\delta t}} \mathcal{L}_{\text{rec}}$ is the vector acquired from minimizing the reconstruction loss.

It is, however, impractical to compute this second term during training, since the reconstruction problem can take hours to solve via optimization. We propose instead to obtain $\mathbf{w}_{t \rightarrow t+\delta t}$ in an online manner, and to jointly optimize frame weights \mathbf{w} over both \mathcal{L}_{rec} and $\mathcal{L}_{\text{pred}}$ at each optimization step. That is, at each step, all current frame weights \mathbf{w} are first used to compute $\mathcal{L}_{\text{pred}}$ and optimize downstream models accordingly, and are then themselves optimized w.r.t. to \mathcal{L}_{rec} . The details of implementing the two losses with batches of frames are introduced in the next section.

4.4 Optimization

During optimization, we sample a short sequence of frames from the training set. For simplifying the notations, we assume that the predictor takes $\tau = 3$ frames as input and predicts the motion of the next frame. An illustration is presented in Fig. 4. Four consecutive frames ($t_i, t_{i+1}, t_{i+2}, t_{i+3}$) are first sampled from the observed sequence and the corresponding embedding vectors $\boldsymbol{\omega}$ are acquired as in Sect. 4.2. Note that training images can be sampled from different synchronized cameras if available.

We disentangle appearance- and motion-related information during optimization by applying \mathcal{L}_{rec} to images reconstructed both with and without motion reparameterization. That is, we sample F for radiance/density values $(\mathbf{c}_t, \boldsymbol{\sigma}_t)$ both as $F(\mathbf{p} + M(\mathbf{p}, \boldsymbol{\omega}_{t \rightarrow t+\delta t}), t + \delta t)$ and as $F(\mathbf{p}, t)$ (*c.f.*, Fig. 4).

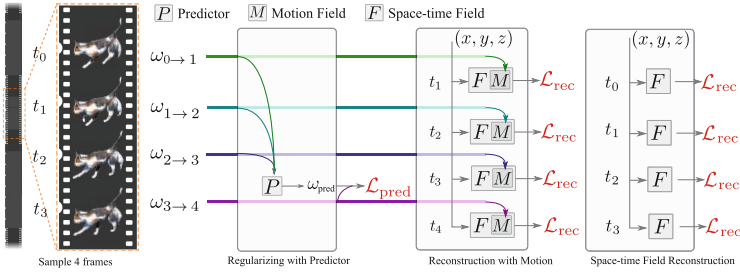


Fig. 4. System optimization, demonstrated on a batch of 4 frames. Predictor P infers a vector ω based on the preceding 3 frames; \mathcal{L}_{pred} minimizes the difference between these predicted embeddings and their sampled equivalents; whereas reconstruction loss \mathcal{L}_{rec} is applied to the predicted four frames, with and without motion reparameterization.

5 Experiments

We qualitatively and quantitatively evaluate our method in this section. *We urge the reader to check our video to better appraise the quality of motion.* The following three datasets are used for evaluation:

- **ZJU-MoCap** [40] is a multi-camera dataset, with videos of one person performing different actions. Since each video sequence records a single human, the scene is less topologically varying and we compare our method with canonical frame-based representations of dynamic scenes. We use videos from 11 cameras for evaluation.
- **Panoptic** [18] includes videos from multiple synchronized cameras under many different settings including multi-person activities and human-object interactions. We select 4 challenging and representative video clips from the 31 HD cameras and denote them as SPORTS, TOOLS, IAN, and CELLO. Each clip has 400 frames and all the clips involve human-object interaction.
- **Hypernerf** [37] is a single-camera dataset, *i.e.*, with one view available at each timestamp. Unlike the previous two datasets that use static cameras, in Hypernerf the multiview information is generated by moving the camera around. Hypernerf is challenging not only because of the single-camera setting, but also the topologically varying scenes.

Details about the clips (*e.g.*, starting and ending frame number) are included in the supplementary. All the sequences are split into short intervals consisting of 25 frames. On each interval, the networks are trained using an Adam optimizer [19] with a learning rate that decays from 5×10^{-4} to 5×10^{-6} every 50k iterations. During training, the two losses are added with a balancing parameter, *i.e.*, $\mathcal{L} = \mathcal{L}_{rec} + \gamma \mathcal{L}_{pred}$ with γ set to 0.01 in all experiments. A batch of 1,024 rays is randomly sampled from the selected frames for training the motion field and the space-time field. We observe that using viewing direction \mathbf{d} in F leads to worse performance if the scene of interest mostly contains Lambertian surfaces. In our experiments, the viewing direction is not taken as the input for the space-time field, *i.e.*, a space-time irradiance field [57]. The network structures of the

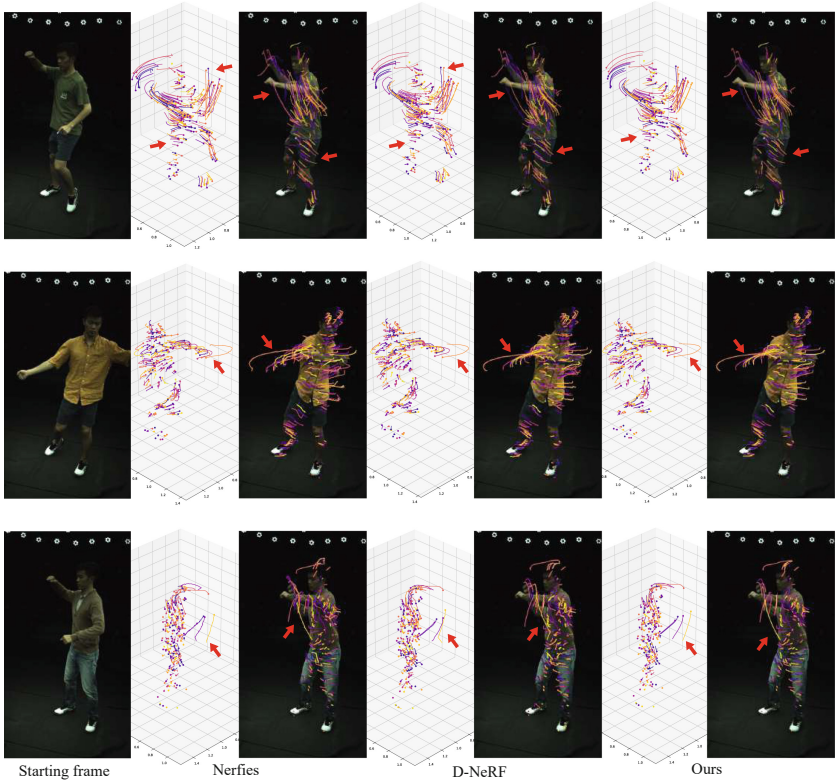


Fig. 5. Comparison of the estimated motion on the ZJU-MoCap dataset. Only one person is captured for each sequence and we compare our method with canonical frame-based methods Nerfies [36] and D-NeRF [41]. Motion for 20 frames is demonstrated.

motion field and the space-time field are the same as in NeRF [30]. The predictor consists of 5 fully connected layers with a width of 128 and ReLU activations.

5.1 Qualitative Evaluation

We visually compare the estimated motion in this section. Since neural motion fields tracks all points in the space, we randomly sample points and then demonstrate their trajectory. Different sampling strategies are used for different datasets. For ZJU-MoCap, we first sample a dense grid of points and then remove the empty points with $\sigma < 20$, then we randomly sample points from the non-empty ones. For Panoptic, since background (walls and floors) is kept in the scenes, we sample meaningful points near the persons in the scene, leveraging provided people positions. For Hypernerf, since the scenes are all front-facing, we sample points on the surfaces according to the depth generated by the space-time field F from one view.

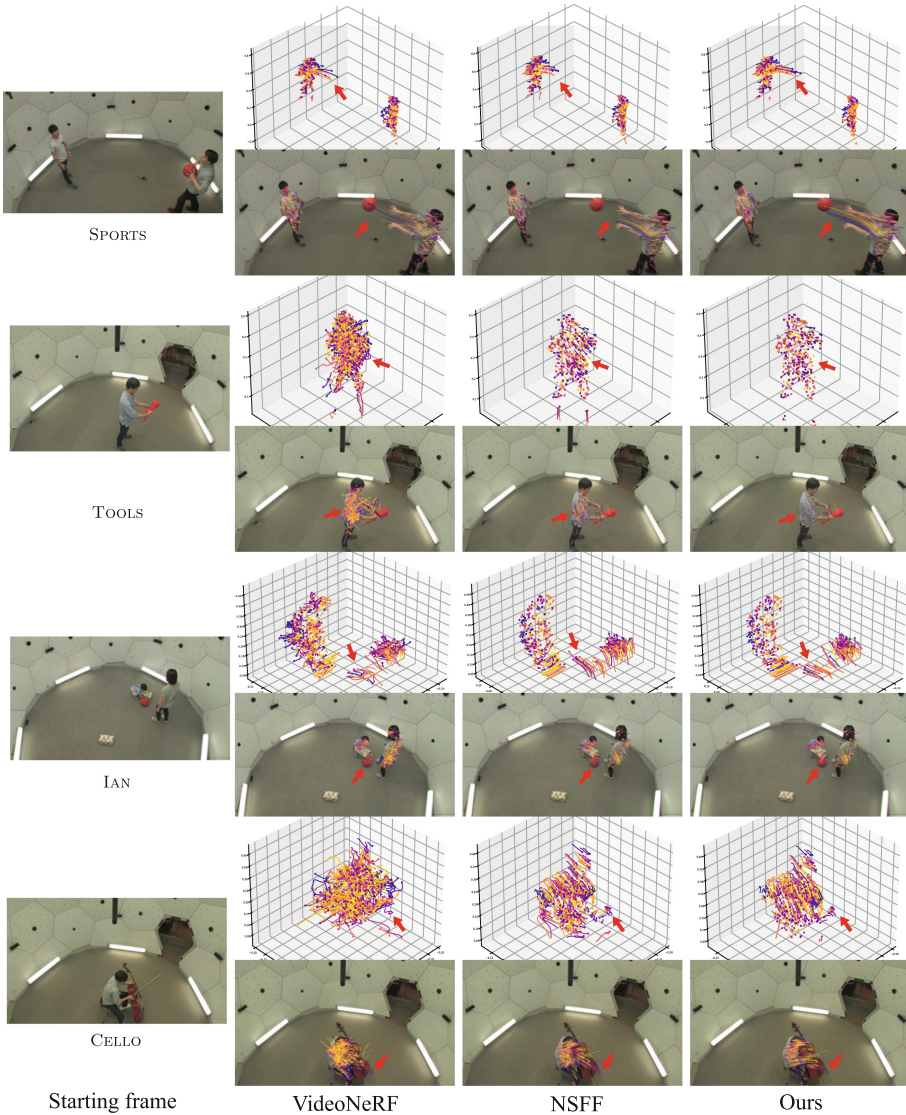


Fig. 6. Motion estimation comparison on the Panoptic dataset [18]. Motions estimated by VideoNeRF are more chaotic than NSFF, possibly due to the 2D optical flow supervision adopted in NSFF. Our method faithfully estimates the motions of people and objects, whereas NSFF fails to track some points, *e.g.*, the ball in SPORTS and IAN.

On Multi-Camera Dataset. We first present our results on ZJU-MoCap in Fig. 5. Since there is only one person in this dataset, the topology of the scene roughly remains unchanged and canonical space-based methods can be applied. Nerfies [36] and D-NerF [41] are selected for comparison. As can be observed

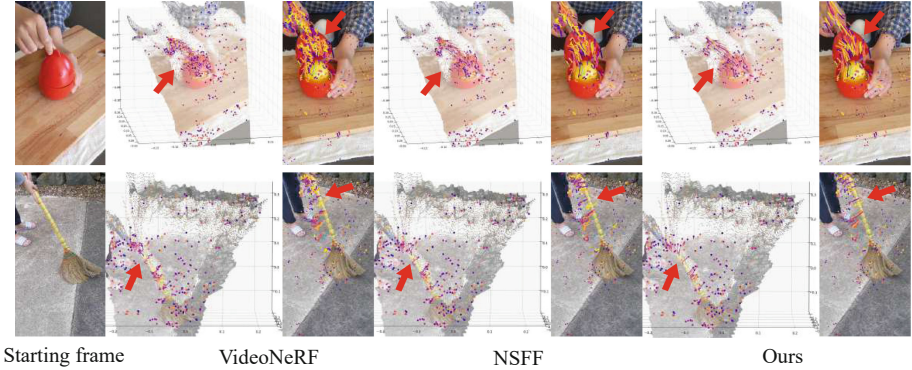


Fig. 7. Comparison of the estimated motion on the Hypernerf dataset [37]. We randomly sample points on the surfaces and then demonstrate their the motions.

from the images, our method can generate a smooth motion as opposed to the rugged and noisy motions from the other two methods.

Figure 6 demonstrate the performance of our method and competitors on the Panoptic dataset. The scenes contain complex geometries and objects may occur or disappear in the middle of a sequence. Two space-time field-based methods, VideoNeRF [57] and NSFF [21], are selected for comparison. Our method estimate the motion of both people and objects accurately, while VideoNeRF presents chaotic results and the motion from NSFF are occasionally inaccurate. The results on Figs. 5 and 6 validate our claim that our method can well track all points in the space without prior knowledge of the scene.

On Single-Camera Dataset. To further validate our method, we demonstrate motion estimation in single-camera settings, which are more commonly encountered by dynamic-scene novel-view rendering methods. We consider the challenging scenes captured by Hypernerf [37]. As shown in Fig. 7, we compare again to VideoNeRF [57] and NSFF [21]. We note that our results are more temporally consistent and accurate than competitors. These results highlight the practical value of our method, able to accurately handle single-camera image sequences captured in the wild.

5.2 Quantitative Evaluation

Quantitative evaluation is difficult for our task since manually labeling a dense set of points in the space is expensive, if not unfeasible. We thus use the sparser human body joints provided by the Panoptic dataset to quantify the accuracy of the estimated motion. MPJPE [45] and 3D-PCK [27] are two widely used metrics for evaluating 3D human pose tracking performance, but both of them do not suit our task since our tracking requires as input the position of points at the starting frame. We propose to calculate the tracking error across K frames

Table 1. Quantitatively evaluating the estimated motion on the Panoptic dataset. Locations of the body joints in the starting frame are used as the inputs and we calculate the averaged tracking error for the body joints.

	mMPJPE ₅ (cm)			mMPJPE ₁₀ (cm)			mMPJPE ₁₅ (cm)		
	VideoNeRF	NSFF	Ours	VideoNeRF	NSFF	Ours	VideoNeRF	NSFF	Ours
SPORTS	5.942	5.171	4.533	8.346	7.933	7.457	11.569	11.254	10.718
TOOLS	3.378	2.341	1.684	4.105	2.879	2.650	4.931	3.984	3.393
IAN	3.448	2.349	2.402	5.059	3.534	3.792	6.767	5.282	4.980
CELLO	2.796	1.759	1.612	4.281	3.296	2.572	4.853	3.776	3.457

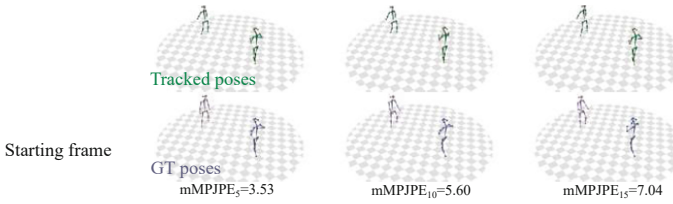


Fig. 8. Visualization of motion tracking results on the Panoptic dataset.

and use the averaged value as a metric. We denote the metric as mMPJPE_K (mean MPJPE), computed as:

$$\text{mMPJPE}_K = \frac{1}{N_f} \frac{1}{K} \sum_{u=1}^{N_f} \sum_{v=u+1}^{u+K} \text{MPJPE}(P_{u \rightarrow v}, P_v^{\text{gt}}), \quad (5)$$

where K is the number of frames for evaluating the motion and N_f is the total number of frames in the sequence. $P_{i \rightarrow j}$ represents the estimated positions for the j th frame given positions for the i th one as inputs, and P_j^{gt} the ground-truth joint positions for the j th frame.

We report the mMPJPE_K metric with $K = 5, 10, 15$ on the Panoptic dataset in Table 1. Our method achieves more accurate tracking performance than the other two methods except on IAN while tracking with 5 and 10 frames. NSFF requires both 2D optical flow and depth, while VideoNeRF requires depth information. As a comparison, we do not use any data-driven prior to guide the motion estimation module. Moreover, in Fig. 8 we visualize the tracked pose and the ground truth pose on one sequence and compute the corresponding mMPJPE metrics ($N_f = 1$ for one sequence).

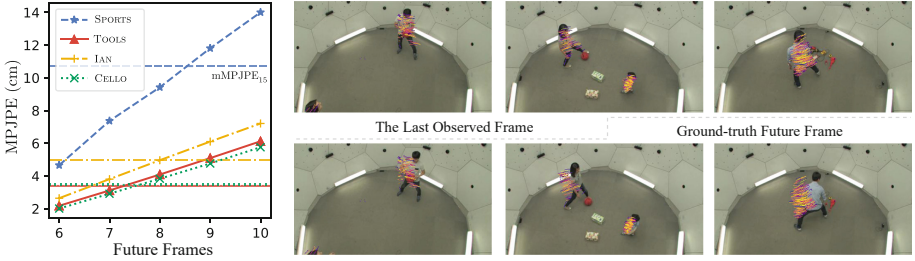


Fig. 9. Accuracy evaluation of the motion predictor. Left: Plotting of the MPJPE of predicted future body joint locations. Horizontal lines are the mMPJPE₁₅ results on the corresponding scenes. Right: Visualization of predicted future motion of densely sampled points on the last observed frames and the 10th ground-truth future frames.

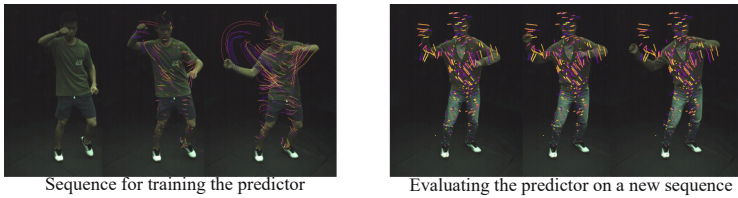


Fig. 10. Transferability of the motion predictor. We train the whole framework on the left-side sequence, then we freeze the predictor and fine tune other models on the right-side sequence. The next 10 frame motions are predicted from the last observed frame (the right-side first image) and visualized. The other two images are real movements in the future 5th frame and 10th frame.

5.3 Analysis of the Motion Predictor

We analyze the motion predictor P in two aspects: prediction accuracy and transferability. For the accuracy evaluation, we compare the predicted future locations of the body joints and the ground-truth future locations. The results are demonstrated in Fig. 9. The training sequences are separated into 20 intervals and we test the prediction results on each interval. The MPJPE of predicted body joint locations are averaged over all the intervals and plotted. We can observe in Table 1 that the model can predict the unseen motion of the next 5 time steps, with a low error close to the tracking error over actual observations.

We further demonstrate the transferability of the predictor in Fig. 10. Since the predictor generates motion codes in a latent space, the same model should work for motion sequences with similar patterns. We test the intuition on the ZJU-MoCap dataset, on two sequences in which the person does similar actions. We can observe from the right side of the figure that the predicted motions align with the real movements. The results demonstrate that the predictor is indeed transferable if the motions are similar.

6 Discussion

Limitations. Our method sometimes fail on non-rigid/monochromatic elements and the problem of motion estimation then gets underconstrained: Some points may converge into the same point for the non-rigid case and it may be hard to tell which part in the monochromatic area moved. We presume that a more advanced (possibly pre-trained) motion prediction model could be leveraged. Moreover, while our methods shows higher precision in estimating natural motion (*e.g.*, dense human motion tracking), it is among our future work to address some other challenging scenes (*e.g.*, scenes with chaotic particles).

Conclusion. We introduced a novel solution for the regularization and prediction of 3D dense motion in dynamic scenes. Leveraging advances in neural fields, we propose a combination of space-time and motion fields conditioned on motion embeddings. Through predictability-based regularization over these embeddings, we promote the encoding of scene-relevant motions and penalize ambiguous and noisy deformations. We acknowledge that this scheme may not benefit all types of scenes (*c.f.* above limitations), but it shows higher precision in natural settings.

References

1. Basha, T., Moses, Y., Kiryati, N.: Multi-view scene flow estimation: a view centered variational approach. *Int. J. Comput. Vision* **101**(1), 6–21 (2013)
2. Boss, M., Braun, R., Jampani, V., Barron, J.T., Liu, C., Lensch, H.: NerD: neural reflectance decomposition from image collections. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12684–12694 (2021)
3. Boss, M., Jampani, V., Braun, R., Liu, C., Barron, J.T., Lensch, H.P.: Neural-pil: neural pre-integrated lighting for reflectance decomposition. *Adv. Neural Inf. Process. Syst.* **34**, 10691–10704 (2021)
4. Box, G.E., Tiao, G.C.: A canonical analysis of multiple time series. *Biometrika* **64**(2), 355–365 (1977)
5. Bozic, A., Palafox, P., Zollhöfer, M., Dai, A., Thies, J., Nießner, M.: Neural non-rigid tracking. *Adv. Neural Inf. Process. Syst.* **33**, 18727–18737 (2020)
6. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: tensorial radiance fields. In: *Proceedings of the European Conference on Computer Vision* (2022)
7. Chibane, J., Pons-Moll, G., et al.: Neural unsigned distance fields for implicit function learning. *Adv. Neural Inf. Process. Syst.* **33**, 21638–21652 (2020)
8. Chung, S.J., Paranjape, A.A., Dames, P., Shen, S., Kumar, V.: A survey on aerial swarm robotics. *IEEE Trans. Rob.* **34**(4), 837–855 (2018)
9. Du, Y., Zhang, Y., Yu, H.X., Tenenbaum, J.B., Wu, J.: Neural radiance flow for 4D view synthesis and video processing. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021)
10. Fang, J., et al.: Fast dynamic radiance fields with time-aware neural voxels. *arXiv preprint [arXiv:2205.15285](https://arxiv.org/abs/2205.15285)* (2022)
11. Gafni, G., Thies, J., Zollhofer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8649–8658 (2021)

12. Goerg, G.: Forecastable component analysis. In: International Conference on Machine Learning, pp. 64–72. PMLR (2013)
13. Hassan, M., et al.: Stochastic scene-aware motion prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 11374–11384 (2021)
14. Hong, Y., Peng, B., Xiao, H., Liu, L., Zhang, J.: Headnerf: a real-time nerf-based parametric head model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20374–20384 (2022)
15. Huang, B., Yan, X., Chen, A., Gao, S., Yu, J.: Pref: phasorial embedding fields for compact neural representations (2022)
16. Huang, T.S., Tsai, R.: Image sequence analysis: motion estimation. In: Huang, T.S. (ed.) Image Sequence Analysis, pp. 1–18. Springer, Heidelberg (1981). https://doi.org/10.1007/978-3-642-87037-8_1
17. Jiang, C., Huang, J., Tagliasacchi, A., Guibas, L.: Shapeflow: learnable deformations among 3D shapes. *Adv. Neural Inf. Process. Syst.* **33**, 9745–9757 (2020)
18. Joo, H., et al.: Panoptic studio: a massively multiview system for social motion capture. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3334–3342 (2015)
19. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) International Conference on Learning Representations (2015)
20. Li, T., et al.: Neural 3D video synthesis from multi-view video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5521–5531 (2022)
21. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
22. Li, Z., Ji, Y., Yang, W., Ye, J., Yu, J.: Robust 3D human motion reconstruction via dynamic template construction. In: International Conference on 3D Vision, pp. 496–505. IEEE (2017)
23. Ling, H.Y., Zinno, F., Cheng, G., Van De Panne, M.: Character controllers using motion vaes. *ACM Trans. Graph. (TOG)* **39**(4), 40–1 (2020)
24. Liu, L., Habermann, M., Rudnev, V., Sarkar, K., Gu, J., Theobalt, C.: Neural actor: neural free-view synthesis of human actors with pose control. *ACM Trans. Graph. (ACM SIGGRAPH Asia)* **40**, 1–16 (2021)
25. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: learning dynamic renderable volumes from images. *ACM Trans. Graph.* **38**(4), 1–14 (2019)
26. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7210–7219 (2021)
27. Mehta, D., et al.: Monocular 3D human pose estimation in the wild using improved cnn supervision. In: 2017 International Conference on 3D Vision (3DV), pp. 506–516. IEEE (2017)
28. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3061–3070 (2015)
29. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: learning 3D reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4460–4470 (2019)

30. 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: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 405–421. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_24
31. Mittal, H., Okorn, B., Held, D.: Just go with the flow: self-supervised scene flow estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11177–11185 (2020)
32. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: reconstruction and tracking of non-rigid scenes in real-time. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 343–352 (2015)
33. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Occupancy flow: 4D reconstruction by learning particle dynamics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5379–5389 (2019)
34. Noguchi, A., Sun, X., Lin, S., Harada, T.: Neural articulated radiance field. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5762–5772 (2021)
35. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 165–174 (2019)
36. Park, K., et al.: Nerfies: deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5865–5874 (2021)
37. Park, K., et al.: Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.* **40**(6) (2021)
38. Pena, D., Box, G.E.: Identifying a simplifying structure in time series. *J. Am. Stat. Assoc.* **82**(399), 836–843 (1987)
39. Peng, S., et al.: Animatable neural radiance fields for modeling dynamic human bodies. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14314–14323 (2021)
40. Peng, S., et al.: Neural body: implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9054–9063 (2021)
41. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10318–10327 (2021)
42. Reddy, N.D., Guigues, L., Pishchulin, L., Eledath, J., Narasimhan, S.G.: Tesse-track: end-to-end learnable multi-person articulated 3D pose tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15190–15200 (2021)
43. Rematas, K., et al.: Urban radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12932–12942 (2022)
44. Schmidt, T., Newcombe, R., Fox, D.: Dart: dense articulated real-time tracking with consumer depth cameras. *Auton. Robots* **39**(3), 239–258 (2015)
45. Sigal, L., Balan, A.O., Black, M.J.: Humaneva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *Int. J. Comput. Vision* **87**(1), 4–27 (2010)
46. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: Nerv: neural reflectance and visibility fields for relighting and view synthesis. In:

- Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7495–7504 (2021)
47. Starke, S., Zhang, H., Komura, T., Saito, J.: Neural state machine for character-scene interactions. *ACM Trans. Graph.* **38**(6), 209–1 (2019)
 48. Stone, J.V.: Blind source separation using temporal predictability. *Neural Comput.* **13**(7), 1559–1574 (2001)
 49. Su, S.Y., Yu, F., Zollhoefer, M., Rhodin, H.: A-nerf: articulated neural radiance fields for learning human shape, appearance, and pose. In: *NeurIPS* (2021)
 50. Tancik, M., et al.: Block-nerf: scalable large scene neural view synthesis. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8248–8258 (2022)
 51. Tewari, A., et al.: State of the art on neural rendering. *Comput. Graph. Forum* **39**(2), 701–727 (2020)
 52. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: reconstruction and novel view synthesis of a dynamic scene from monocular video. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12959–12970 (2021)
 53. Turki, H., Ramanan, D., Satyanarayanan, M.: Mega-nerf: scalable construction of large-scale nerfs for virtual fly-throughs. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12922–12931 (2022)
 54. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 722–729. IEEE (1999)
 55. Vogel, C., Schindler, K., Roth, S.: Piecewise rigid scene flow. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1377–1384 (2013)
 56. Wang, C., Eckart, B., Lucey, S., Gallo, O.: Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint [arXiv:2105.05994](https://arxiv.org/abs/2105.05994)* (2021)
 57. Xian, W., Huang, J.B., Kopf, J., Kim, C.: Space-time neural irradiance fields for free-viewpoint video. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9421–9431 (2021)
 58. Xiangli, Y., et al.: Citynerf: building nerf at city scale. *arXiv preprint [arXiv:2112.05504](https://arxiv.org/abs/2112.05504)* (2021)
 59. Xie, Y., et al.: Neural fields in visual computing and beyond. In: *Computer Graphics Forum*, vol. 41, pp. 641–676. Wiley Online Library (2022)
 60. Yang, B., et al.: Learning object-compositional neural radiance field for editable scene rendering. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13779–13788 (2021)
 61. Yoon, J.S., Kim, K., Gallo, O., Park, H.S., Kautz, J.: Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5336–5345 (2020)
 62. Zhai, M., Xiang, X., Lv, N., Kong, X.: Optical flow and scene flow estimation: a survey. *Pattern Recogn.* **114**, 107861 (2021)
 63. Zhang, J., et al.: Editable free-viewpoint video using a layered neural representation. *ACM Trans. Graph.* **40**(4), 149:1–149:18 (2021)