Factorized Motion Fields for Fast Sparse Input Dynamic View Synthesis

Nagabhushan Somraj
Indian Institute of Science
Bengaluru, Karnataka, India
nagabhushans@iisc.ac.in

Kapil Choudhary
Indian Institute of Science
Bengaluru, India
kapilc@iisc.ac.in

Sai Harsha Mupparaju
Indian Institute of Science
Bengaluru, India
saiharsham@iisc.ac.in

Rajiv Soundararajan
Indian Institute of Science
Bengaluru, India
rajivs@iisc.ac.in

ABSTRACT

Designing a 3D representation of a dynamic scene for fast optimization and rendering is a challenging task. While recent explicit representations enable fast learning and rendering of dynamic radiance fields, they require a dense set of input viewpoints. In this work, we focus on learning a fast representation for dynamic radiance fields with sparse input viewpoints. However, the optimization with sparse input is under-constrained and necessitates the use of motion priors to constrain the learning. Existing fast dynamic scene models do not explicitly model the motion, making them difficult to be constrained with motion priors. We design an explicit motion model as a factorized 4D representation that is fast and can exploit the spatio-temporal correlation of the motion field. We then introduce reliable flow priors including a combination of sparse flow priors across cameras and dense flow priors within cameras to regularize our motion model. Our model is fast, compact and achieves very good performance on popular multi-view dynamic scene datasets with sparse input viewpoints. The source code for our model can be found on our project page: https://nagabhushansn95.github.io/publications/2024/RF-DeRF.html.

CCS CONCEPTS

• **Computing methodologies** → **Rendering**; **Volumetric models**; *Computer vision*; *Computational photography*; *Regularization*.

KEYWORDS

Fast dynamic view synthesis, dynamic radiance fields, factorized models, motion priors, sparse input views

1 INTRODUCTION

Synthesizing novel views of a scene given images/videos of the scene in other viewpoints has several applications such as metaverse, sports, and remote exploration. Neural Radiance Fields (NeRF) [Mildenhall et al. 2020] brought in a seminal shift in novel view synthesis by incorporating differential volume rendering with a compact continuous-depth model. However, it has several limitations such as the inability to handle object motion in the scene as well as long training and rendering times. Recently, K-Planes [Fridovich-Keil et al. 2023] significantly reduced the time complexity for dynamic view synthesis by proposing a factorized 4D volume representation. However, K-Planes requires a large number of input viewpoints to render photo-realistic novel views when employing a multi-view camera setup. The goal of our work is to overcome this limitation and design a fast dynamic radiance field that can effectively learn the dynamic scene with few input viewpoints. We mainly focus on a sparse multi view camera setting, where a video from each viewpoint is available.

Optimizing a dynamic radiance field with few input viewpoints is highly under-constrained. A popular approach to regularize an under-constrained system is to impose additional priors during the optimization. For example, sparse static radiance fields are regularized with depth [Deng et al. 2022; Uy et al. 2023], visibility [Somraj and Soundararajan 2023] and natural image [Wu et al. 2024] priors. However, the object motion in dynamic scenes motivates the study of motion priors for sparse input dynamic view synthesis. Since K-Planes employs a 4D volumetric representation without a motion model, it does not allow the motion implicitly learned by the model to be regularized using motion priors. Thus, there is a need to design a dynamic radiance field with an explicit motion field that lends itself to be constrained with motion supervision.

We design a dynamic radiance field consisting of two models, a 5D radiance field that learns the 3D scene at a canonical time instant and a 4D motion or deformation field that learns the motion from any time instant t to the canonical time instant t'. Since the motion model is a one-directional mapping from t to t', it is not obvious how to impose flow priors across two arbitrary time instants. We achieve this by constraining our motion model to map a pair of matched points, obtained using the motion prior, to the same 3D point in the canonical volume. Fig. 2 illustrates our approach of imposing the motion prior. This allows us to impose motion priors across any two time instants and across any cameras.

Prior and concurrent works [Fang et al. 2022; Shaw et al. 2023] employ deep neural networks (DNN) to learn the motion field. However, the use of the DNN to learn the motion makes the model computationally expensive. The challenge here is to design a motion model that can learn the motion efficiently while yielding fast training and rendering. While explicit models such as voxel grids [Fridovich-Keil et al. 2022; Sun et al. 2022], hash-grids [Müller et al. 2022] and 3D Gaussians (3DGS) [Kerbl et al. 2023] are shown to be effective in learning static scenes, naively extending these techniques to model 4D motion may not be efficient. For example, extending 3D voxel grids to 4D scales the memory requirement to the fourth power of grid resolution. Further, for a given object, since the motion exists at every time instant, models that exploit scene sparsity to reduce the memory requirement may not extend effectively to 4D. Since the motion is dense and has a high spatio-temporal correlation, we employ factorized volumes to exploit the correlation. Specifically, we employ a 4D factorized volume to learn the scene flow from any time t to the canonical time t'.

An ideal motion prior to regularize the motion field is a dense flow prior across cameras and time instants. Prior models such as NSFF [Li et al. 2021] use dense optical flow to supervise an auxiliary task of predicting the flow between two frames, which may not efficiently regularize the core dynamic radiance field model. Further, we observe that deep flow estimation networks used in such works are sensitive to variations in camera parameters, lighting changes and occlusions, and hence find it challenging to determine reliable dense correspondences across cameras. Regularizing the model with such noisy flow priors may lead to sub-optimal performance.

We address these challenges by employing a combination of two complementary flow priors, a rich but sparse flow prior across cameras and a dense flow prior within individual cameras. In particular, we obtain reliable flow priors across cameras and time instants for a sparse set of SIFT keypoints. Within individual cameras, we obtain reliable dense flow priors using deep optical flow estimation networks [Teed and Deng 2020]. These priors within the same camera tend to be more reliable owing to smaller variations in camera parameters and lighting changes. The reliability of our priors as opposed to the dense flow priors across cameras is illustrated in Fig. 3. The combination of the sparse across camera flow priors and the dense within camera flow priors provides an effective alternative to the cross camera dense flow priors.

We evaluate our model on two popular multi-view dynamic scene datasets and find that our model outperforms the state-of-the-art dynamic view synthesis models with fewer input viewpoints. We refer to our model as RF-DeRF, since we employ Reliable Flow priors for Deformable Radiance Fields. We summarize the main contributions of our work in the following:

- We design a fast and compact dynamic radiance field for sparse input dynamic view synthesis by employing an explicit motion model that can be easily regularized using motion priors. We employ a 4D factorized volume to exploit the spatio-temporal correlation of the motion field.
- We propose a complimentary set of reliable flow priors to regularize the motion field when training with few multi-view videos. We obtain a sparse flow prior that regularizes

	Dynamic Scenes	Sparse Views	Explicit Model	Motion Model
NeRF	✗	✗	✗	–
TensoRF, i-ngp, 3DGS	✗	✗	✓	–
DS-NeRF, SimpleNeRF	✗	✓	✗	–
NSFF*, DyNeRF	✓	✗	✗	✗
D-NeRF	✓	✗	✗	✓
TiNeuVox, SWAGS	✓	✗	✗†	✓
K-Planes, HexPlane	✓	✗	✓	✗
Ours	✓	✓	✓	✓

Table 1: Related work overview: We compare our work with prior works based on various aspects. Sparse input views refers to models that can handle data from few stationary multi-view cameras. Explicit refers to models that primarily employ explicit models followed by an optional tiny MLP. † TiNeuVox and SWAGS use an explicit model to represent the scene, but use implicit neural networks to model the motion. * NSFF predicts motion as an auxiliary task.

flow across cameras and a dense flow prior that regularizes the flow within individual cameras.
- We achieve very good dynamic view synthesis performance on two popular multi-view datasets with very few views.

2 RELATED WORK

2.1 Classical Work on Deformation Models

The modeling of dynamic radiance fields through a static radiance field and a motion field is similar to the use of a deformation model that deforms a canonical representation of an entity. This is a popular approach to model deformable solids [Sederberg and Parry 1986], human motion [Loper et al. 2015], facial expressions [Bradley et al. 2010], deformable garments [Miguel et al. 2012; Pons-Moll et al. 2017], fluids [Adrian 1991; Gregson et al. 2012; Hawkins et al. 2005], gases [Atcheson et al. 2008], smoke [Hawkins et al. 2005], flames [Ihrke and Magnor 2004] as well as CT (Computed Tomography) and MRI (Magnetic Resonance Imaging) scans in the medical field [Nunes et al. 2012]. Learning the motion or deformation field is also found to improve the reconstruction of 3D scenes [Vedula et al. 2000] and free-viewpoint rendering of dynamic scenes [Carranza et al. 2003].

The canonical space in such deformable models is popularly represented as triangular meshes estimated through a multi-view stereo algorithm [Allen et al. 2003; Bradley et al. 2010, 2008; Kraevoy and Sheffer 2005], or specialized parametric meshes for humans and faces [Anguelov et al. 2005; Loper et al. 2015; Vlasic et al. 2008], or sums of Gaussians [Stoll et al. 2011]. In contrast, we employ a radiance field to learn the canonical space owing to its advantages over meshes [Mildenhall et al. 2020]. Different deformation models include free-form deformation (FFD) fields or optical flows [Faloutsos et al. 1997; Wang et al. 2009], bijective mappings [Kraevoy and Sheffer 2004], morphable models [Blanz and Vetter 1999], splines [Vlasic et al. 2009], volumetric Laplacian deformations [de Aguiar et al. 2008] and linear systems with basis functions [Ihrke and Magnor 2004]. Our motion field is closer to the free-form deformation among

the above deformation approaches, but differs in the representation used for the motion field. Finally, the sparse flow priors we employ can be thought of as similar to tracking the markers [Guskov et al. 2003; Hasler et al. 2006] or keypoints [Bradley et al. 2008; Joo et al. 2018] while estimating the classical deformation models.

2.2 Dynamic View Synthesis

Different from the use of deformable models, Zitnick et al. [2004] learn the 3D scene dynamics using a layered depth representation with motion compensation. More recent volumetric representations such as Multiplane Images (MPI) [Zhou et al. 2018] are extended to handle dynamic scenes by employing an MPI per frame [Lin et al. 2021] or temporal basis functions [Xing and Chen 2021]. However, such approaches suffer from depth discretization artifacts [Mildenhall et al. 2020]. Instead of volumetric models, Yoon et al. [2020] employ depth image based warping, where the depth is obtained by combining monodepth with multi-view depth. Recent work on dynamic view synthesis with sparse input viewpoints require depth information obtained through the use of RGB-D cameras [Li et al. 2023] or multi-view stereo [Bansal and Zollhöfer 2023]. Further, such approaches struggle to handle soft edges and translucent objects [Penner and Zhang 2017].

2.3 Dynamic Radiance Fields

In contrast to the classical approaches, dynamic view synthesis can be solved by learning a 6D radiance field that maps the position, time and viewing direction to the radiance. However, the performance of such approaches degrades when the input viewpoints are sparse. We provide a quick overview of prior work that share a few attributes of our model in Tab. 1. To the best of our knowledge, ours is the first work to address dynamic radiance fields with few input viewpoints while achieving fast training and rendering. In the following, we review the literature on dynamic radiance fields and sparse-input static radiance fields.

Dynamic radiance fields can be broadly classified into two categories based on how the temporal modeling is handled, which is crucial in the sparse input setting. A simple approach is to model the dynamic radiance field as a 6D function of position, time, and viewing direction. To utilize flow priors, NSFF [Li et al. 2021] predicts flow as an auxiliary task and constrains the volume density of the mapped 3D points provided by the flow. K-Planes [Fridovich-Keil et al. 2023] and HexPlane [Cao and Johnson 2023] extend TensoRF [Chen et al. 2022] to a 4D model that maps the position and time to a latent feature, which is then decoded by a tiny multi-layer perceptron (MLP). Instead of conditioning the radiance field directly on time, DyNeRF [Li et al. 2022] conditions the radiance field on a per-time-instant feature vector which is jointly optimized with the radiance field. The lack of a motion model in these approaches makes it incompatible to impose motion priors when learning with sparse input viewpoints.

The second set of models employs a motion or deformation field that maps the 3D points from a given time instant to a canonical time instant [Pumarola et al. 2021; Wang et al. 2023b,a]. TiNeuVox [Fang et al. 2022], SWAGS [Shaw et al. 2023] and CoGS [Yu et al. 2023] replace the scene representation MLP in D-NeRF [Pumarola et al. 2021] with a TensoRF or 3DGS model, but use MLPs to model the

motion field, leading to expensive training and rendering time. To achieve fast optimization and rendering, DeVRF [Liu et al. 2022] employs a 4D voxel grid to learn the deformation field, but its memory requirements scale with the fourth power of the grid resolution. Instead of employing a free-form deformation model, Wang et al. [2021] models the motion using discrete cosine transform (DCT) basis functions. Different from the above, prior [Guo et al. 2023] and concurrent work [Wu et al. 2023] employ a motion model that maps the 3D points from the canonical time instant to a given time instant. However, the forward warping of points can lead to holes in novel views, which may need to be infilled using a separate inpainting network [Guo et al. 2023; Mirzaei et al. 2023; Somraj et al. 2022].

2.4 Radiance Fields with Sparse Views

Optimizing a radiance field with sparse input views for static scenes has rich literature. With sparse inputs, the volume rendering equations are severely under-constrained leading to distortions such as floaters. Few-shot NeRF models address this by constraining the model further with hand-crafted or learned priors. While the early works employ simple priors such as sparsity of mass [Kim et al. 2022] and smoothness of depth [Niemeyer et al. 2022], later works found that the use of richer priors can lead to better performance. The most popular prior is the depth prior [Deng et al. 2022; Roessle et al. 2022; Somraj et al. 2023; Uy et al. 2023], while other priors based on visibility [Somraj and Soundararajan 2023], semantic information [Jain et al. 2021] and diffusion models [Wu et al. 2024; Wynn and Turmukhambetov 2023] are also explored. While most of the works employ the NeRF baseline leading to long training time, recent works explore the use of explicit models [Shi et al. 2024; Xiong et al. 2023; Zhu et al. 2023]. However, none of these approaches analyze or regularize the distortions caused by motion in a dynamic scene.

2.5 Fast Optimization of Radiance Fields

Fast optimization of radiance fields allows wider usage in real-world applications. Voxel grid based approaches trade memory for speed by replacing the MLP in NeRF with a voxel grid [Fridovich-Keil et al. 2022; Sun et al. 2022]. Prior work on fast and compact representation for learning radiance fields can be broadly classified into two categories. One category of models employs a factorized volume representation [Cao and Johnson 2023; Chen et al. 2022; Fridovich-Keil et al. 2023] to exploit the spatial correlation of the entities to be learned, such as volume density and color. The other category of models [Barron et al. 2023; Kerbl et al. 2023; Müller et al. 2022] mainly exploit the sparsity of the scene to reduce the memory footprint. However, the scene flow for a moving object is non-zero at every time instant, but its variation is temporally correlated. Since the motion field is not sparse temporally, we believe that the sparsification based approaches may tend to learn an independent motion field for every time instant. For example, with a 4DGS model, a single 4D Gaussian may not be able to model the motion across a few time instants and can model the motion at a single time instant only, similar to Lee et al. [2023]. On the other hand, factorized volumes can effectively exploit the spatio-temporal correlation of
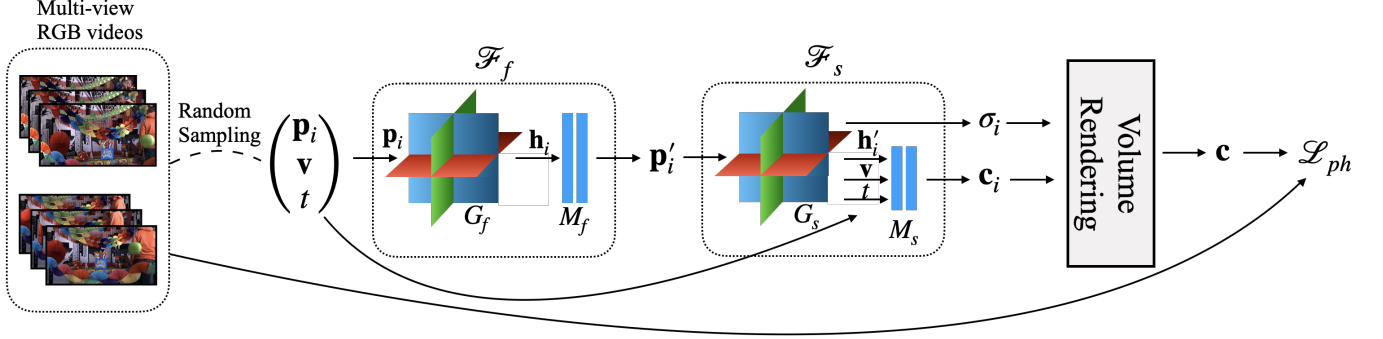
Figure 1: Model architecture: We decompose the dynamic radiance field into a 4D scene flow or deformation field $\mathcal{F}_f$ that maps a 3D point p_i at time t to the corresponding 3D point p'_i at canonical time t', and a 5D radiance field $\mathcal{F}_s$ that models the scene at canonical time t'. Both the fields are modeled using a factorized volume followed by a tiny MLP, which allows fast optimization and rendering. We note that G_f is modeled using six planes, although we show only three owing to the difficulty in visualizing four dimensions. The MLP M_s is conditioned on time and viewing direction to model time-dependent color variations such as shadows and view-dependent color variations such as specularities. The output of $\mathcal{F}_s$ is volume rendered to obtain the color of the pixel and the photometric loss is used to train both the fields. The explicitly modeled motion field $\mathcal{F}_f$ is additionally regularized using the flow priors as shown in Fig. 2.

the motion field and hence we employ a 4D factorized model to learn the motion field.

3 METHOD

Given a set of posed video frames I_t^v, where $t = \{1, 2, \ldots, N_f\}$ denotes the time instant or the frame index among N_f frames per camera and $v = \{1, 2, \ldots N_c\}$ denotes the camera index among N_c stationary cameras, the goal is to synthesize the dynamic scene at a specified time instant $t = \{1, 2, \ldots, N_f\}$ and in any novel view. We focus on the sparse multi-view setting, where the number of cameras N_c is small. The main challenges here include the design of an explicit motion model that allows regularization with motion priors and the choice of reliable motion priors itself when training with few input views. We first describe the design of our dynamic radiance field (Sec. 3.1), and then discuss its training with reliable flow priors (Sec. 3.2).

3.1 Dynamic Radiance Field

We modify the K-Planes [Fridovich-Keil et al. 2023] implementation for dynamic radiance fields to incorporate an explicit motion model. K-Planes employs a multi-resolution grid $\mathbf{G}$ followed by a tiny multi-layer perceptron (MLP) $\mathbf{M}$ to represent a radiance field. As shown in Fig. 1, we model our dynamic radiance field by employing two factorized tensorial models, a radiance field $\mathcal{F}_s = \mathbf{G}_s \circ \mathbf{M}_s$ that represents the 3D scene at a canonical time instant t', and a 4D scene flow or deformation field $\mathcal{F}_f = \mathbf{G}_f \circ \mathbf{M}_f$ that represents the scene flow for a 3D point $\mathbf{p}$ from time t to t'. Mapping the 3D scene at every time instant to a canonical volume helps our model enforce temporal consistency of objects in the scene [Pumarola et al. 2021].

Here, $\mathbf{G}_f$ consists of six planes $\{\mathbf{S}_{xy}, \mathbf{S}_{yz}, \mathbf{S}_{xz}, \mathbf{S}_{xt}, \mathbf{S}_{yt}, \mathbf{S}_{zt}\}$ at every resolution, where the first three planes model the spatial correlation and the next three model the spatio-temporal correlation of the motion field. To obtain the scene flow for a 3D point $\mathbf{p}$ from time t to t', we first project the 4D point $(\mathbf{p}, t)$ onto the six planes

and bilinearly interpolate the feature vectors in each of the six planes. We then combine the features using the Hadamard product to obtain the final feature vector $\mathbf{h}_i$ as

$$\mathbf{h}_i = \mathbf{G}_f(\mathbf{p}) = \prod_{c \in \{xy, yz, xz, xt, yt, zt\}} S_c(\mathbf{p}), \qquad (1)$$

which is then fed to the tiny MLP $\mathbf{M}_f$ that outputs the scene flow. Modeling the scene flow field $\mathcal{F}_f$ using the hex-plane representation and the tiny MLP makes our motion model fast and compact. For more details on the hex-plane representation, we refer the readers to K-Planes [Fridovich-Keil et al. 2023]. We model the scene grid $\mathbf{G}_s$ using a similar factorization with three spatial planes. While it is beneficial to model the motion field using a factorized volume, the canonical scene can be learned using any explicit representation such as 3DGS [Kerbl et al. 2023]. We employ factorized volumes to model both scene and motion using a unified framework.

We train our model, similar to K-Planes, by rendering randomly sampled pixels $\mathbf{q}_t^v$ in view v at time t and using the ground truth color as supervision. To render a pixel $\mathbf{q}_t^v$, we sample N_p points $\{\mathbf{p}_i\}_{i=1}^{N_p}$ at depths $\{z_i\}_{i=1}^{N_p}$ along the corresponding ray. For every 3D point $\mathbf{p}_i$, we first obtain the corresponding 3D point $\mathbf{p}'_i$ at canonical time t' by computing the scene flow from time t to t' using $\mathcal{F}_f$ as

$$\mathbf{p}'_i = \mathcal{F}_f(\mathbf{p}_i, t) + \mathbf{p}_i. \qquad (2)$$

We then query $\mathbf{G}_s$ at $\mathbf{p}'_i$ to obtain the volume density σ_i and a latent feature $\mathbf{h}'_i$ corresponding to $\mathbf{p}_i$ as

$$\sigma_i, \mathbf{h}'_i = \mathbf{G}_s(\mathbf{p}'_i). \qquad (3)$$

A tiny MLP $\mathbf{M}_s$ maps $\mathbf{h}'_i$, encoded viewing direction $\mathbf{v}$ and encoded time t to the color $\mathbf{c}_i$ of $\mathbf{p}_i$ as

$$\mathbf{c}_i = \mathbf{M}_s(\mathbf{h}'_i, \gamma(\mathbf{v}), \gamma(t)), \qquad (4)$$

where γ denotes the encoding [Müller et al. 2022] of the viewing direction and the time instant. Conditioning $\mathbf{M}_s$ additionally on
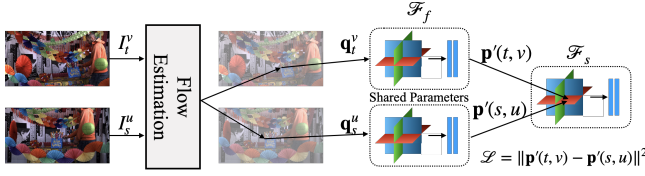
Figure 2: Flow regularization: Since the motion field $\mathcal{F}_f$ gives only the unidirectional flow from time t to t', we impose the flow prior by minimizing the distance between the 3D points in the canonical volume corresponding to the matched pixels $(\mathbf{q}_t^v, \mathbf{q}_s^u)$ in the input frames (I_t^v, I_s^u).

time allows our model to capture illumination changes due to object motion. $\mathbf{c}_i$ are then volume rendered to obtain the color $\mathbf{c}$ of $\mathbf{q}$ as

$$w_i = \left(\Pi_{j=1}^{i-1} \exp\left(-\delta_j \sigma_j\right) \right) \left(1 - \exp\left(-\delta_i \sigma_i\right) \right), \quad (5)$$

$$\mathbf{c} = \sum_{i=1}^{N_p} w_i \mathbf{c}_i, \quad z = \sum_{i=1}^{N_p} w_i z_i, \quad (6)$$

where $\delta_i = z_i - z_{i-1}$ and z gives the expected depth of $\mathbf{q}_t^v$. Both $\mathcal{F}_f$ and $\mathcal{F}_s$ are optimized through the photometric loss, $\mathcal{L}_{ph} = \|\mathbf{c} - \hat{\mathbf{c}}\|^2$, where $\hat{\mathbf{c}}$ is the ground truth color. Decomposing the scene into a canonical scene field and a flow field allows us to regularize the flow field using flow priors when only a sparse set of viewpoints are available.

3.2 Training with Flow Priors

A dense flow prior across cameras and across time instants is an ideal prior for optimal regularization of our motion field. However, different cameras in a multi-view setting could suffer from variations in parameters such as intrinsics, color balance, exposure and focus making it challenging to determine correspondences across cameras based on luminance constancy. As a result, the dense flow estimates obtained using deep optical flow networks are unreliable across cameras as seen in Fig. 3a. On the other hand, matching keypoints using robust SIFT [Lowe 2004] descriptors are more reliable across cameras as seen in Fig. 3b. Specifically, for a pair of frames across time instants t and s and cameras v and u, we extract the SIFT keypoints in individual images using Colmap [Schonberger and Frahm 2016] and match the keypoints using SIFT descriptors. The difference between the locations of the matched keypoints gives the sparse flow prior $\mathcal{P}_{sf}$. While Colmap provides a reasonable number of matches, newer models such as R2D2 [Revaud et al. 2019] could be employed to improve the richness of the sparse flow prior.

Employing flow priors from any time t to canonical time t' for large $t - t'$ may be less efficient since large regions of the scene may not be visible. Thus, it is desirable to utilize flow priors across a short duration of time. However, the motion model $\mathcal{F}_f$ provides the scene flow from t to t' only and not the flow between any two arbitrary time instants. We resolve this by using the flow prior to encourage $\mathcal{F}_f$ to map a pair of matched points at different time instants to the same object location in 3D. Specifically, we obtain the matching pixels using the flow priors and constrain the 3D points corresponding to the matching pixels to map to the same 3D point in the canonical scene field as shown in Fig. 2.

(a) Dense flow across cameras estimated by RAFT.

(b) Sparse flow across cameras estimated by SIFT.
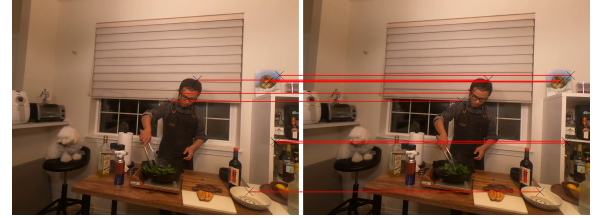
(c) Dense flow within cameras estimated by RAFT.

Figure 3: Visualization of different flow priors: We show the matched pixels as provided by different flow priors. The pixels in the first view are randomly picked from those for which sparse flow is available and the same pixels are used for dense flow. Note that the second view in the first two examples has more blur as compared to the first view. (a) We show that the dense flow priors across cameras obtained using deep optical flow networks such as RAFT [Teed and Deng 2020] are prone to erroneous matches, due to variations in camera parameters and lighting. We observed similar trends with other deep optical flow networks as well such as AR-Flow [Liu et al. 2020]. (b) Matching pixels across cameras using robust SIFT features provides reliable matches, albeit sparse. (c) Within individual cameras, the dense correspondences provided by deep optical flow networks are more reliable owing to smaller variations in lighting.

Mathematically, let the flow corresponding to a pixel $\mathbf{q}_t^v$ to time s and view u be given by $\mathbf{f}_{t \to s}^{v \to u}$. Then the matching pixel at time s and view u is given by $\mathbf{q}_s^u = \mathbf{q}_t^v + \mathbf{f}_{t \to s}^{v \to u}$. Let the 3D points sampled along the rays corresponding to $\mathbf{q}_t^v$ and $\mathbf{q}_s^u$ be given by $\mathbf{p}_i(t, v)$ and $\mathbf{p}_i(s, u)$ respectively. Then, we impose the sparse flow constraint as

$$\mathcal{L}_{sf} = \left\| \sum_{i=0}^{N_p - 1} w_i(t, v) \mathbf{p}_i'(t, v) - \sum_{i=0}^{N_p - 1} w_i(s, u) \mathbf{p}_i'(s, u) \right\|^2, \quad (7)$$

where $\mathbf{p}'(t, v)$ and $\mathbf{p}'(s, u)$ are computed from $\mathbf{p}(t, v)$ and $\mathbf{p}(s, u)$ using $\mathcal{F}_f$ as in Eq. (2), and $w_i(t, v)$ and $w_i(s, u)$ are computed using Eq. (5). The two terms in Eq. (7) represent $\mathbf{q}_t^v$ and $\mathbf{q}_s^u$ in the canonical

		N3DV				InterDigital			
	Model	PSNR ↑	SSIM ↑	LPIPS ↓	Depth MAE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Depth MAE ↓
2 views	HexPlane	13.81	0.41	0.58	2.03	13.78	0.24	0.55	1.51
	K-Planes	17.95	0.62	0.43	0.59	16.51	0.51	0.40	0.35
	DeRF	17.98	0.61	0.44	0.67	17.04	0.56	0.37	0.35
	RF-DeRF	**20.35**	**0.71**	**0.33**	**0.41**	**19.08**	**0.66**	**0.31**	**0.23**
3 views	HexPlane	15.53	0.49	0.50	1.97	13.85	0.27	0.54	1.43
	K-Planes	23.65	0.83	0.25	0.34	18.75	0.66	0.30	0.22
	DeRF	22.61	0.81	0.27	0.39	18.72	0.68	0.30	0.22
	RF-DeRF	**25.07**	**0.88**	**0.21**	**0.16**	**20.41**	**0.74**	**0.25**	**0.12**
4 views	HexPlane	15.50	0.47	0.54	1.95	15.18	0.38	0.49	1.24
	K-Planes	24.80	0.86	0.20	0.27	**23.85**	**0.86**	**0.16**	0.11
	DeRF	23.76	0.83	0.24	0.36	23.51	0.85	0.17	0.13
	RF-DeRF	**25.86**	**0.89**	**0.19**	**0.17**	23.69	**0.86**	0.17	**0.10**

Table 2: Quantitative results: We compare our model with K-Planes on N3DV [Li et al. 2022] and InterDigital [Sabater et al. 2017] datasets with two, three, and four input views. We also show the performance of our base DeRF model for reference. We report the PSNR, SSIM, and LPIPS scores for the rendered images and the depth MAE for the rendered depth maps. The best scores in each category are shown in bold.

volume respectively. Thus, $\mathcal{L}_{sf}$ guides $\mathcal{F}_s$ on which two points in two different time instants belong to the same object.

In Eq. (7), we first find the canonical field points $\mathbf{p}'_i$ using $\mathcal{F}_f$ and then average the location of the 3D points using weights w_i and not the other way for the following reason. The former approach regularizes the motion model $\mathcal{F}_f$ for every $\mathbf{p}_i$ giving a rich supervision to $\mathcal{F}_s$, whereas the latter approach regularizes $\mathcal{F}_f$ only for the expected 3D point $\mathbf{p} = \sum w_i \mathbf{p}_i$. Further, since we do not impose stop-gradient on w_i in $\mathcal{L}_{sf}$, the flow priors also help remove incorrect masses such as floaters.

While the sparse flow prior $\mathcal{P}_{sf}$ is rich on account of matching pixels across cameras and time instants, the prior is available at only sparse keypoints. Thus, we complement $\mathcal{P}_{sf}$ with a dense flow prior $\mathcal{P}_{df}$ within individual cameras. For a pair of frames from the same camera at two closer time instants, there is little variation in the camera intrinsics and pipeline parameters. Hence, the dense flow priors obtained using a deep optical flow network are more reliable as seen in Fig. 3c. We employ the popular optical flow estimation model RAFT [Teed and Deng 2020] to obtain the dense flow priors and constrain our model with a dense flow loss $\mathcal{L}_{df}$ similar to Eq. (7). We find that the dense flow priors typically help reduce artifacts in static regions.

We train our model by minimizing the combination of photometric loss $\mathcal{L}_{ph}$, sparse flow loss $\mathcal{L}_{sf}$ and dense flow loss $\mathcal{L}_{df}$ as

$$\mathcal{L} = \mathcal{L}_{ph} + \lambda_{sf}\mathcal{L}_{sf} + \lambda_{df}\mathcal{L}_{df}, \tag{8}$$

where λ_{sf} and λ_{df} are hyper-parameters.

4 EXPERIMENTS

4.1 Evaluation Setup

Datasets: We evaluate our model on two popular multi-view dynamic scene datasets, namely N3DV [Li et al. 2022] and InterDigital [Sabater et al. 2017] with two, three, and four input views. Following prior work [Fridovich-Keil et al. 2023], we downsample the videos spatially by a factor of two for all the experiments.

Figure 4: Qualitative examples on N3DV dataset with two input views: We can observe that K-Planes finds it hard to learn the moving person leading to significant distortions. Our DeRF model (without any priors) corrects a few errors by virtue of the common canonical volume. Imposing our priors leads to much better reconstruction.

We use the video at the center of the camera rig for testing and uniformly sample train videos from the remaining videos. *N3DV dataset* contains six real-world scenes with 17–21 static cameras per scene and 300 frames per viewpoint. The videos have a spatial resolution of 1352×1014 and a frame rate of 30fps. *InterDigital dataset* contains multiple real-world scenes with 16 static cameras and varying number of frames per scene. We undistort the video frames using the radial distortion parameters provided with the dataset and use the undistorted videos for our experiments. We select five scenes that contain at least 300 frames and choose the first 300 frames. The videos have a spatial resolution of 1024×544 and a frame rate of 30fps spanning 10 seconds in all our experiments.

Evaluation measures: We evaluate the rendered frames of all the methods using PSNR, SSIM [Wang et al. 2003] and LPIPS [Zhang et al. 2018]. We also evaluate the models on their ability to reconstruct the 3D scene by computing MAE on the rendered depth maps. Due to the unavailability of true depth maps on both datasets, we use the depth provided by K-Planes trained with dense input views as pseudo ground truth depth.

$\mathcal{P}_{\text{sf}}$	$\mathcal{P}_{\text{df}}$	PSNR ↑	SSIM ↑	LPIPS ↓	Depth MAE ↓
		22.61	0.81	0.27	0.39
✓		24.79	0.87	0.22	0.20
	✓	24.10	0.85	0.23	0.31
✓	✓	**25.07**	**0.88**	**0.21**	**0.16**

Table 3: Quantitative results of ablations on N3DV dataset with three input views: We evaluate the significance of the two flow priors we employ, namely the sparse flow prior ($\mathcal{P}_{\text{sf}}$) and the dense flow prior ($\mathcal{P}_{\text{df}}$) by disabling one component at a time. We also show the performance of the DeRF model without any priors for reference.

	N3DV		InterDigital	
Model	LPIPS ↓	Depth MAE ↓	LPIPS ↓	Depth MAE ↓
w/o any priors	0.27	0.39	0.30	0.22
w/ cross-camera dense flow	0.30	0.49	0.36	0.29
w/ our flow priors	**0.21**	**0.16**	**0.25**	**0.12**

Table 4: We test the performance of our DeRF model with dense flow priors across cameras and our reliable flow priors. The performance of the DeRF model (without any priors) is also shown for reference.

4.2 Comparisons and Implementation Details

We mainly compare the performance of our model against K-Planes and HexPlane. We use the official code released by the authors of K-Planes and modify it to implement our model. We refer to our base model without any priors as DeRF model. To test the superiority of our priors, we also compare our model against K-Planes with sparse depth priors and DeRF with dense flow priors across cameras. Since we use two factorized models instead of one used in K-Planes, we reduce the feature dimension in both of our models by half to keep the total number of parameters comparable to K-Planes. We train all the models for 30k iterations on a single NVIDIA RTX 2080 GPU. We randomly pick $s \in \{t - 10, t + 10\}$ to impose the flow prior losses and set $\lambda_{\text{sf}} = 1$ and $\lambda_{\text{df}} = 1$. We use the same values as suggested by K-Planes for all the remaining hyperparameters.

4.3 Results

We show the quantitative performance of our model, K-Planes and HexPlane in Tab. 2, where we observe that our RF-DeRF model outperforms both K-Planes and HexPlane across all the settings on both the datasets. We observe that the performance of all the models is relatively higher on the N3DV dataset as compared to the InterDigital dataset. This is perhaps due to the InterDigital dataset having larger motion and highly textured regions. We also note that the performance of HexPlane is substantially lower than both K-planes and our model. This could be a result of optimizing Hex-Plane initially with a low-resolution grid, which causes the model to overfit the input-views and not utilize the high-resolution grid in the later stages of training. K-Planes does not suffer from this drawback by using a multi-resolution grid throughout the optimization. From Figs. 6, 7 and 9, we observe that our model is able to correct

	N3DV		InterDigital	
Model	SSIM ↑	Depth MAE ↓	SSIM ↑	Depth MAE ↓
K-Planes	0.83	0.34	0.66	0.22
K-Planes + $\mathcal{P}_{\text{sd}}$	0.82	0.26	0.71	0.20
DeRF	0.81	0.39	0.68	0.22
DeRF + $\mathcal{P}_{\text{sd}}$	0.80	0.34	0.70	0.22
DeRF + $\mathcal{P}_{\text{sf}}$	0.87	0.20	0.73	0.13
RF-DeRF	**0.88**	**0.16**	**0.74**	**0.12**

Table 5: We analyze the impact of our sparse flow prior that regularizes both the scene geometry and the learned motion against the sparse depth prior [Deng et al. 2022] that regularizes only the scene geometry. The models are trained with three input views on both the datasets.

errors both in moving objects and in static regions over K-Planes. In particular, we observe that K-Planes suffer from blur, deformation, and disappearance of moving objects while rendering novel views, which are mitigated by our model. The improvements are more starkly visible in the supplementary videos. We compare the rendered depth from both the models in Figs. 10 and 11 and observe that our model is able to reconstruct the depth more accurately than K-Planes. Tab. 2 also shows that our DeRF model (without any priors) performs better than K-Planes in the scenes with larger motion, such as the scenes in the InterDigital dataset, perhaps due to the temporal consistency enforced by the canonical volume. This is also visible in the example shown in Fig. 4. In our experiments, RF-DeRF roughly took 1.5 hours and 5GB GPU memory to train on a single scene, and 6 seconds to render a single frame. The size of the saved model parameters is approximately 280MB. Please refer to Fridovich-Keil et al. [2023] for the training time for various models including K-Planes and DyNeRF.

Ablations: We analyze the significance of each component of our model by disabling one component at a time in Tab. 3. We observe that the sparse flow prior is the most significant component of our model, giving a large boost in performance when included and a large drop in performance when excluded. The effect of sparse flow prior is more pronounced in terms of depth MAE, which measures the accuracy of the reconstructed 3D scene. This demonstrates the importance of reliable motion priors across cameras, even if sparse. Fig. 8 shows that sparse flow prior is more helpful in mitigating motion related artifacts such as motion blur, while within-camera dense flow prior mainly helps reduce distortions in the static regions. The combination of the two priors leads to the best performance by mitigating both artifacts.

Dense cross-camera flow priors vs our priors: To validate our hypothesis that a combination of cross-camera sparse flow and within-camera dense flow priors is more effective than dense flow priors across cameras, we impose cross-camera dense flow priors on our base DeRF model and evaluate the performance on both datasets. From Tab. 4 and Fig. 12, we observe that imposing noisy cross-camera dense flow priors leads to a large drop in performance, whereas our priors help improve the performance significantly.

Depth vs flow priors: We now analyze the effect of flow priors against that of depth priors in dynamic view synthesis with sparse input views. The depth priors guide the model on the exact locations
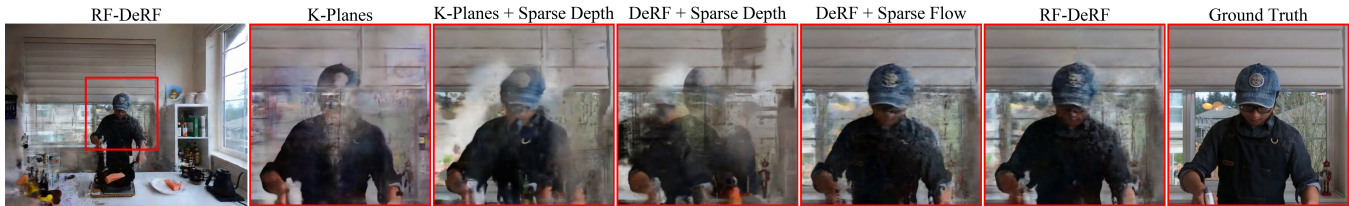
Figure 5: Qualitative examples for depth priors vs flow priors: We observe that sparse depth prior is not very effective in improving the reconstruction quality. However, our sparse flow prior is highly effective in mitigating the distortions, while the use of sparse and dense flow priors in our final model leads to the best reconstruction quality.

Model	N3DV			InterDigital		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
K-Planes	30.55	0.96	0.12	29.00	0.96	0.07
DeRF	29.95	0.95	0.12	28.14	0.95	0.09

Table 6: We test the performance of our DeRF model against K-Planes with dense input views. Please see the videos in the supplementary for visual comparisons.

of the objects in the scene, but do not account for the correlation of object locations across time instants. On the other hand, the flow priors account for such inter-frame correlations as well as encompass the information provided by the depth priors in static regions. To experimentally validate our hypothesis, we compare the performance of K-Planes and our DeRF models when regularized with sparse depth priors $\mathcal{P}_{sd}$ obtained through Colmap sparse reconstruction [Deng et al. 2022]. We obtain $\mathcal{P}_{sd}$ for every frame and use them to constrain the K-Planes model as $\mathcal{L}_{sd} = \|z - \hat{z}\|^2$, where z is the depth obtained in Eq. (6) and $\hat{z}$ is the depth prior. From Tab. 5, we observe that sparse flow priors lead to a significant improvement in performance as compared to that of sparse depth priors. The difference in performance can be more clearly observed in Fig. 5 and the supplementary videos. This shows the utility of motion priors in dynamic view synthesis with sparse input views.

Performance with increasing viewpoints: To understand the impact of our priors as the number of input views increases, we compare the performance of RF-DeRF against K-Planes with different number of cameras on a single scene from N3DV dataset in Fig. 13. Our model significantly improves the performance over K-Planes in sparse viewpoint scenarios, while matching the performance of K-Planes with dense input views. Thus, we conclude that our priors are particularly helpful in improving the reconstruction of under-observed regions in the scene.

Performance with dense input views: We now test the ability of our model in reconstructing the dynamic scene when provided with dense multi-view inputs. Since motion priors may not be needed when dense views are available, we analyze the performance of our base DeRF model (without any priors). From Tab. 6, we observe that our DeRF model is competitive with K-Planes. Thus, one could employ our model in both dense and sparse input scenarios by additionally employing reliable flow priors in the latter case.

4.4 Limitations and Future Work

Our approach of mapping the scene at every time instant to the canonical volume implies that only the objects present in the canonical volume can be rendered. In other words, our model may not handle objects entering or leaving the scene, which could happen over longer durations. This could be resolved by learning multiple models, each trained on self-contained shorter duration videos. We employ Colmap to generate sparse flow priors in our framework, which is a time-consuming process. In our experiments, we found that the generation of sparse flow priors takes about 45 minutes per scene. There is a need to explore faster approaches to determine reliable sparse correspondences across cameras. Further, our approach requires the cameras to be calibrated and time-synchronized apriori. It would be interesting to optimize the scene, camera parameters, and time-synchronization jointly in a single framework.

5 CONCLUSION

We consider the setting of fast dynamic view synthesis when only a few videos of the scene as observed from different static cameras are available. By exploiting the spatio-temporal correlation of the motion field, we design an explicit motion model using factorized representations that is compact, fast, and allows effective regularization with the flow priors. We observe that the naive use of cross-camera dense flow priors has a negative effect on the performance, while a careful imposition of reliable cross-camera sparse and within-camera dense flow priors provides a significant boost in performance. Further, we also show that the motion priors are more general and encompass the depth priors in dynamic view synthesis with sparse input views. We demonstrate the effectiveness of our approach on two popular datasets and show that our approach outperforms the state-of-the-art fast and compact dynamic radiance fields by a large margin when only a few viewpoints are available.

ACKNOWLEDGMENTS

REFERENCES

Ronald J Adrian. 1991. Particle-Imaging Techniques for Experimental Fluid Mechanics. *Annual review of fluid mechanics* 23, 1 (1991), 261–304.

Brett Allen, Brian Curless, and Zoran Popović. 2003. The Space of Human Body Shapes: Reconstruction and Parameterization from Range Scans. *ACM Transactions on Graphics (TOG)* 22, 3 (July 2003), 587–594. https://doi.org/10.1145/882262.882311

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: Shape Completion and Animation of People. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. https://doi.org/10.1145/1186822.1073207

Bradley Atcheson, Ivo Ihrke, Wolfgang Heidrich, Art Tevs, Derek Bradley, Marcus Magnor, and Hans-Peter Seidel. 2008. Time-resolved 3D Capture of Non-stationary Gas Flows. *ACM Transactions on Graphics (TOG)* 27, 5 (December 2008). https://doi.org/10.1145/1409060.1409085

Aayush Bansal and Michael Zollhöfer. 2023. Neural Pixel Composition for 3D-4D View Synthesis From Multi-Views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2023. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. https://doi.org/10.1145/311535.311556

Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. 2010. High Resolution Passive Facial Performance Capture. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. https://doi.org/10.1145/1833349.1778778

Derek Bradley, Tiberiu Popa, Alla Sheffer, Wolfgang Heidrich, and Tamy Boubekeur. 2008. Markerless Garment Capture. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. https://doi.org/10.1145/1399504.1360698

Ang Cao and Justin Johnson. 2023. HexPlane: A Fast Representation for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. 2003. Free-Viewpoint Video of Human Actors. *ACM Transactions on Graphics (TOG)* 22, 3 (July 2003), 569–577. https://doi.org/10.1145/882262.882309

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. TensoRF: Tensorial Radiance Fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. 2008. Performance Capture from Sparse Multi-View Video. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. https://doi.org/10.1145/1399504.1360697

Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. 2022. Depth-Supervised NeRF: Fewer Views and Faster Training for Free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

P. Faloutsos, M. Van de Panne, and D. Terzopoulos. 1997. Dynamic Free-Form Deformations for Animation Synthesis. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 3, 3 (1997), 201–214. https://doi.org/10.1109/2945.620488

Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. 2022. Fast Dynamic Radiance Fields with Time-Aware Neural Voxels. In *Proceedings of the SIGGRAPH Asia 2022 Conference Papers*. https://doi.org/10.1145/3550469.3555383

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. 2023. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields Without Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

James Gregson, Michael Krimerman, Matthias B Hullin, and Wolfgang Heidrich. 2012. Stochastic Tomography and its Applications in 3D Imaging of Mixing Fluids. *ACM Transactions on Graphics (TOG)* 31, 4 (July 2012), 1–10.

Xiang Guo, Jiadai Sun, Yuchao Dai, Guanying Chen, Xiaoqing Ye, Xiao Tan, Errui Ding, Yumeng Zhang, and Jingdong Wang. 2023. Forward Flow for Novel View Synthesis of Dynamic Scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Igor Guskov, Sergey Klibanov, and Benjamin Bryant. 2003. Trackable Surfaces. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*.

Nils Hasler, Mark Asbach, Bodo Rosenhahn, Jens-Rainer Ohm, and Hans-Peter Seidel. 2006. Physically based Tracking of Cloth. In *Proceedings of the International Workshop on Vision, Modeling, and Visualization (VMV)*.

Tim Hawkins, Per Einarsson, and Paul Debevec. 2005. Acquisition of Time-Varying Participating Media. *ACM Transactions on Graphics (TOG)* 24, 3 (July 2005), 812–815. https://doi.org/10.1145/1073204.1073266

Ivo Ihrke and Marcus Magnor. 2004. Image-Based Tomographic Reconstruction of Flames. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*. https://doi.org/10.1145/1028523.1028572

Ajay Jain, Matthew Tancik, and Pieter Abbeel. 2021. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Hanbyul Joo, Tomas Simon, and Yaser Sheikh. 2018. Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (TOG)* 42, 4 (2023).

Mijeong Kim, Seonguk Seo, and Bohyung Han. 2022. InfoNeRF: Ray Entropy Minimization for Few-Shot Neural Volume Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vladislav Kraevoy and Alla Sheffer. 2004. Cross-Parameterization and Compatible Remeshing of 3D Models. *ACM Transactions on Graphics (TOG)* 23, 3 (August 2004), 861–869. https://doi.org/10.1145/1015706.1015811

Vladislav Kraevoy and Alla Sheffer. 2005. Template-Based Mesh Completion. In *Proceedings of the Symposium on Geometry Processing (SGP)*.

Yao-Chih Lee, Zhoutong Zhang, Kevin Blackburn-Matzen, Simon Niklaus, Jianming Zhang, Jia-Bin Huang, and Feng Liu. 2023. Fast View Synthesis of Casual Videos. *arXiv e-prints*, Article arXiv:2312.02135 (2023), arXiv:2312.02135 pages. arXiv:2312.02135

Deqi Li, Shi-Sheng Huang, Tianyu Shen, and Hua Huang. 2023. Dynamic View Synthesis with Spatio-Temporal Feature Warping from Sparse Views. In *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*. https://doi.org/10.1145/3581783.3612419

Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. 2022. Neural 3D Video Synthesis From Multi-View Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 2021. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kai-En Lin, Lei Xiao, Feng Liu, Guowei Yang, and Ravi Ramamoorthi. 2021. Deep 3D Mask Volume for View Synthesis of Dynamic Scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2022. DeVRF: Fast Deformable Voxel Radiance Fields for Dynamic Scenes. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. 2020. Learning by Analogy: Reliable Supervision From Transformations for Unsupervised Optical Flow Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Transactions on Graphics (TOG)* 34, 6 (October 2015). https://doi.org/10.1145/2816795.2818013

David G Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)* 60 (2004), 91–110.

Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A Otaduy, and Steve Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. *Computer Graphics Forum (CGF)* 31, 2pt2 (2012), 519–528. https://doi.org/10.1111/j.1467-8659.2012.03031.x

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G. Derpanis, Jonathan Kelly, Marcus A. Brubaker, Igor Gilitschenski, and Alex Levinshtein. 2023. SPIn-NeRF: Multiview Segmentation and Perceptual Inpainting With Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–15.

Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. 2022. RegNeRF: Regularizing Neural Radiance Fields for View Synthesis From Sparse Inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hilario Nunes, Yurdagul Uzunhan, Thomas Gille, Christine Lamberto, Dominique Valeyre, and Pierre-Yves Brillet. 2012. Imaging of Sarcoidosis of the Airways and Lung Parenchyma and Correlation with Lung Function. *European Respiratory Journal* 40, 3 (2012), 750–765. https://doi.org/10.1183/09031936.00025212

Eric Penner and Li Zhang. 2017. Soft 3D Reconstruction for View Synthesis. *ACM Transactions on Graphics (TOG)* 36, 6 (November 2017), 1–11. https://doi.org/10.1145/3130800.3130855

Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. 2017. ClothCap: Seamless 4D Clothing Capture and Retargeting. *ACM Transactions on Graphics (TOG)* 36, 4 (July 2017). https://doi.org/10.1145/3072959.3073711

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. 2019. R2D2: Reliable and Repeatable Detector and Descriptor. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. 2022. Dense Depth Priors for Neural Radiance Fields From Sparse Input Views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Neus Sabater, Guillaume Boisson, Benoit Vandame, Paul Kerbiriou, Frederic Babon, Matthieu Hog, Remy Gendrot, Tristan Langlois, Olivier Bureller, Arno Schubert, and Valerie Allie. 2017. Dataset and Pipeline for Multi-View Light-Field Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*.

Johannes L. Schonberger and Jan-Michael Frahm. 2016. Structure-From-Motion Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thomas W. Sederberg and Scott R. Parry. 1986. Free-form Deformation of Solid Geometric Models. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. https://doi.org/10.1145/15922.15903

Richard Shaw, Jifei Song, Arthur Moreau, Michal Nazarczuk, Sibi Catley-Chandar, Helisa Dhamo, and Eduardo Perez-Pellitero. 2023. SWAGS: Sampling Windows Adaptively for Dynamic 3D Gaussian Splatting. *arXiv e-prints*, Article arXiv:2312.13308 (2023), arXiv:2312.13308 pages. arXiv:2312.13308

Ruoxi Shi, Xinyue Wei, Cheng Wang, and Hao Su. 2024. ZeroRF: Fast Sparse View 360° Reconstruction with Zero Pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nagabhushan Somraj, Adithyan Karanayil, and Rajiv Soundararajan. 2023. SimpleNeRF: Regularizing Sparse Input Neural Radiance Fields with Simpler Solutions. In *Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques - Asia (SIGGRAPH-Asia)*.

Nagabhushan Somraj, Pranali Sancheti, and Rajiv Soundararajan. 2022. Temporal View Synthesis of Dynamic Scenes through 3D Object Motion Estimation with Multi-Plane Images. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. https://doi.org/10.1109/ISMAR55827.2022.00100

Nagabhushan Somraj and Rajiv Soundararajan. 2023. ViP-NeRF: Visibility Prior for Sparse Input Neural Radiance Fields. In *Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH)*. https://doi.org/10.1145/3588432.3591539

Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. 2011. Fast Articulated Motion Tracking using a Sums of Gaussians Body Model. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2011.6126338

Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022. Direct Voxel Grid Optimization: Super-Fast Convergence for Radiance Fields Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zachary Teed and Jia Deng. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Mikaela Angelina Uy, Ricardo Martin-Brualla, Leonidas Guibas, and Ke Li. 2023. SCADE: NeRFs from Space Carving with Ambiguity-Aware Depth Estimates. (June 2023).

S. Vedula, S. Baker, S. Seitz, and T. Kanade. 2000. Shape and Motion Carving in 6D. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2000.854926

Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. 2008. Articulated Mesh Animation from Multi-View Silhouettes. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. https://doi.org/10.1145/1399504.1360696

Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. 2009. Dynamic Shape Capture using Multi-View Photometric Stereo. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. https://doi.org/10.1145/1661412.1618520

Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. 2021. Neural Trajectory Fields for Dynamic Novel View Synthesis. *arXiv e-prints*, Article arXiv:2105.05994 (2021), arXiv:2105.05994 pages. arXiv:2105.05994

Chaoyang Wang, Lachlan Ewen MacDonald, László A. Jeni, and Simon Lucey. 2023b. Flow Supervision for Deformable NeRF. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huamin Wang, Miao Liao, Qing Zhang, Ruigang Yang, and Greg Turk. 2009. Physically Guided Liquid Surface Modeling from Videos. *ACM Transactions on Graphics (TOG)* 28, 3 (July 2009). https://doi.org/10.1145/1531326.1531396

Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. 2023a. Tracking Everything Everywhere All at Once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *Proceedings of the Asilomar Conference on Signals, Systems Computers*.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Wang Xinggang. 2023. 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. *arXiv e-prints*, Article arXiv:2310.08528 (2023), arXiv:2310.08528 pages. arXiv:2310.08528

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. 2024. ReconFusion: 3D Reconstruction with Diffusion Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jamie Wynn and Daniyar Turmukhambetov. 2023. DiffusioNeRF: Regularizing Neural Radiance Fields with Denoising Diffusion Models. *arXiv e-prints*, Article arXiv:2302.12231 (2023), arXiv:2302.12231 pages. arXiv:2302.12231

Wenpeng Xing and Jie Chen. 2021. Temporal-MPI: Enabling Multi-Plane Images for Dynamic Scene Modelling via Temporal Basis Learning. *arXiv e-prints*, Article arXiv:2111.10533 (2021), arXiv:2111.10533 pages. arXiv:2111.10533

Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. 2023. SparseGS: Real-Time 360° Sparse View Synthesis using Gaussian Splatting. *arXiv e-prints*, Article arXiv:2312.00206 (2023), arXiv:2312.00206 pages. arXiv:2312.00206

Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. 2020. Novel View Synthesis of Dynamic Scenes With Globally Coherent Depths From a Monocular Camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Heng Yu, Joel Julin, Zoltán Á Milacski, Koichiro Niinuma, and László A Jeni. 2023. CoGS: Controllable Gaussian Splatting. *arXiv e-prints*, Article arXiv:2312.05664 (2023), arXiv:2312.05664 pages. arXiv:2312.05664

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo Magnification: Learning View Synthesis Using Multiplane Images. *ACM Transactions on Graphics (TOG)* 37, 4 (July 2018).

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. 2023. FSGS: Real-Time Few-shot View Synthesis using Gaussian Splatting. *arXiv e-prints*, Article arXiv:2312.00451 (2023), arXiv:2312.00451 pages. arXiv:2312.00451

C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. 2004. High-Quality Video View Interpolation using a Layered Representation. *ACM Transactions on Graphics (TOG)* 23, 3 (August 2004), 600–608. https://doi.org/10.1145/1015706.1015766
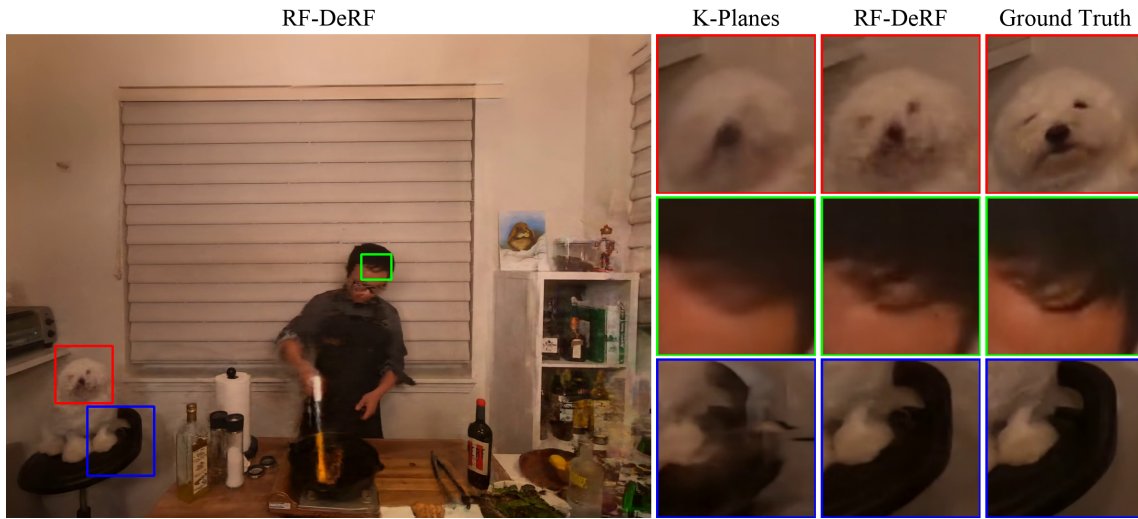
Figure 6: Qualitative examples on N3DV dataset with three input views: We observe that K-Planes blurs moving objects such as the face of the dog and the hairs on the face of the person in the first two examples respectively. We also find that K-Planes creates distortions in the static regions when an object moves in its vicinity as seen in the third row. However, our RF-DeRF model produces sharper and more accurate results.
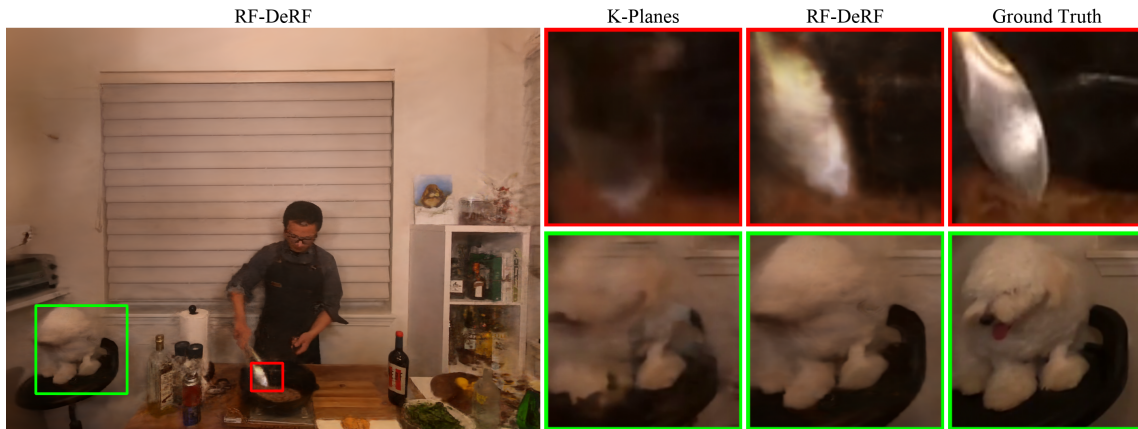
Figure 7: Qualitative examples on N3DV dataset with four input views: We find that K-Planes is unable to learn the moving tongs in the first row, while it struggles to learn the shape of the dog in the second row. Our model is able to reconstruct both, albeit with some blur.
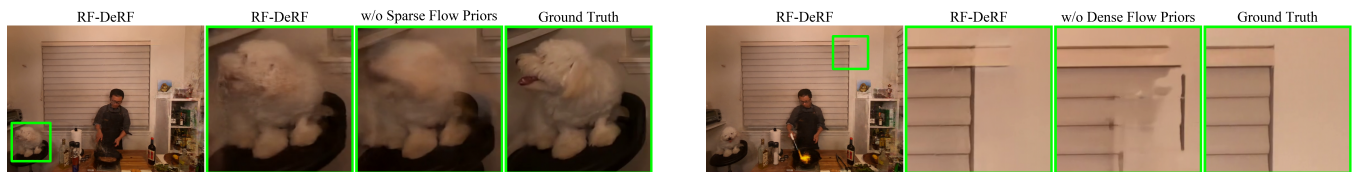
(a) Without sparse flow priors, we observe that the face of the dog suffers from motion blur creating a fuzzy white mass.

(b) Absence of dense flow priors leads to distortions on the wall, where we observe the window curtain bleeding onto the wall.

Figure 8: Qualitative examples of ablated models on N3DV dataset: Sparse flow prior is effective in regularizing moving regions in the scene. On the other hand, dense flow prior is effective in regularizing the smooth and static regions. Employing both the priors leads to the best reconstruction without either artifacts.

Nagabhushan Somraj, Kapil Choudhary, Sai Harsha Mupparaju, and Rajiv Soundararajan
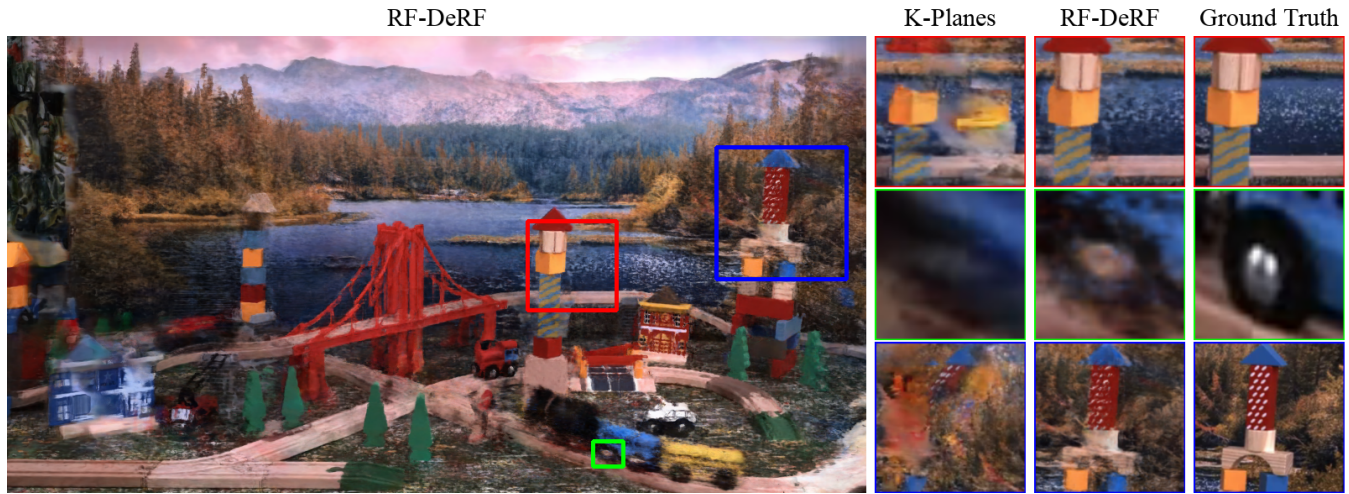
Figure 9: Qualitative examples on InterDigital dataset: In the first row, K-Planes creates a duplication of the tower, while the tower is significantly distorted in the third row. Our model correctly reconstructs both the towers and better reconstructs the wheel of the moving train.
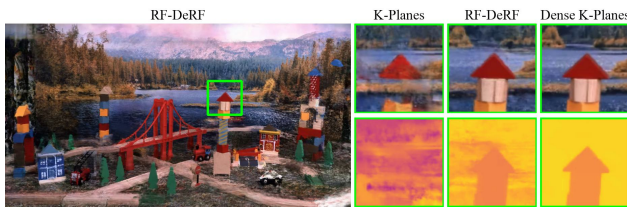
Figure 10: Visualization of rendered depth on InterDigital dataset: While our model learns better depth in the scene leading to better reconstruction of the tower, K-Planes is unable to learn the geometry correctly causing distortions in the tower.
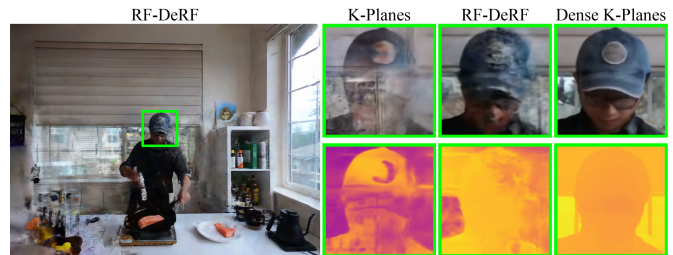
Figure 11: Visualization of rendered depth on N3DV dataset: Observe the difference in color of the depth map rendered by K-Planes which shows the errors in the estimated depth.
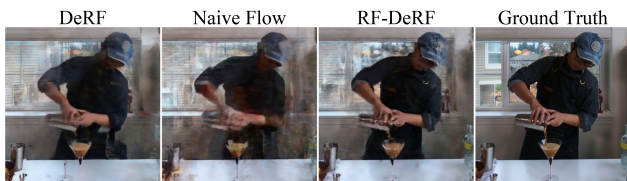
Figure 12: Qualitative examples to show the effect of naive cross-camera dense flow priors: We observe that the hand of the person in the second column is distorted, perhaps due to incorrect priors provided by the cross-camera dense flow. Further, we observe that the cross-camera dense flow priors deteriorate the performance as compared to our base DeRF model. Our priors do not cause such distortions, while improving the overall reconstruction quality (observe the color of the shirt).
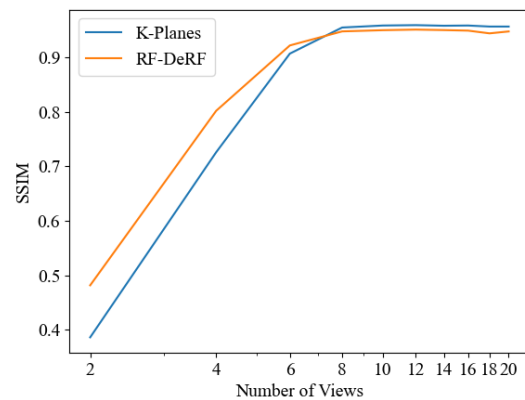
Figure 13: Quantitative results on the flame salmon scene of N3DV dataset with different number of input views: We observe that our model outperforms K-Planes in sparse input setting, while being competitive when more input views are available.

Supplement

The contents of this supplement include

- A. Video examples on N3DV and InterDigital datasets.
- B. Extensive quantitative evaluation reports.

A VIDEO COMPARISONS

We compare our ReFlow-DyRF model with K-Planes [Fridovich-Keil et al. 2023], ablated models, and various other baselines we design. We divide the video comparisons into three sets. In the first set, we compare our model against the vanilla K-Planes and the different baselines we design by incorporating sparse-depth and cross-camera dense flow priors. In the second set, we show the effect of removing each of our priors. Finally, we compare our base DyRF without any priors against K-Planes when dense input views are available. The videos are available on our project website https://nagabhushansn95.github.io/publications/2024/RF-DeRF.html.

B PERFORMANCE ON INDIVIDUAL SCENES

For the benefit of follow-up work, where researchers may want to analyze the performance of different models or compare the models on individual scenes, we provide the performance of various models on individual scenes in Tabs. 7 to 12.

Table 7: Per-scene performance of various models with three input views on N3DV dataset. The four rows show PSNR, SSIM, LPIPS Alex, and Depth MAE scores, respectively.

Model \ Scene Name	Coffee Martini	Cook Spinach	Cut Roasted Beef	Flame Salmon	Flame Steak	Sear Steak	Average
K-Planes	20.11	24.24	26.55	16.92	26.77	27.29	23.65
	0.7590	0.8743	0.9128	0.5975	0.9178	0.9216	0.8305
	0.3215	0.2059	0.1767	0.4717	0.1541	0.1520	0.2470
	0.4501	0.3275	0.1570	0.7002	0.2178	0.1718	0.3374
K-Planes + Sparse Depth	20.66	22.70	26.66	17.19	24.52	25.02	22.79
	0.7830	0.8557	0.9090	0.6167	0.8767	0.8971	0.8230
	0.2672	0.1916	**0.1541**	0.4149	0.1700	**0.1486**	0.2244
	0.2910	0.2468	**0.1058**	0.5459	0.2011	0.1640	0.2591
DyRF	19.82	23.15	25.19	16.93	25.33	25.25	22.61
	0.7531	0.8361	0.8946	0.5852	0.8819	0.8805	0.8052
	0.3282	0.2338	0.2018	0.4579	0.1917	0.1909	0.2674
	0.4619	0.3320	0.2080	0.6320	0.3483	0.3340	0.3860
DyRF + Sparse Depth	19.81	22.64	25.29	16.47	24.59	24.85	22.27
	0.7741	0.8316	0.8813	0.5572	0.8676	0.8783	0.7983
	0.3088	0.2503	0.2105	0.4732	0.2199	0.1983	0.2768
	0.3694	0.3332	0.1473	0.6096	0.3325	0.2484	0.3401
DyRF + Cross-Cam Dense Flow	19.52	23.95	24.94	17.73	25.24	25.24	22.77
	0.6990	0.8439	0.8718	0.6043	0.8776	0.8832	0.7966
	0.4035	0.2443	0.2402	0.4929	0.2055	0.1990	0.2976
	0.6330	0.4498	0.3664	0.7207	0.3959	0.3937	0.4932
ReFlow-DyRF w/o Sparse Flow	20.58	25.61	26.79	17.93	27.07	26.62	24.10
	0.8016	**0.9052**	0.9174	0.6680	0.9233	0.9099	0.8542
	0.2923	**0.1808**	0.1835	0.3944	**0.1533**	0.1687	0.2288
	0.3089	0.2936	0.2293	0.5329	0.2365	0.2569	0.3097
ReFlow-DyRF w/o Dense Flow	21.10	26.03	**27.55**	19.56	26.85	**27.66**	24.79
	0.8106	0.9003	**0.9225**	0.7452	0.9153	**0.9280**	0.8703
	0.2863	0.1875	0.1760	0.3622	0.1671	0.1518	0.2218
	0.2648	0.1791	0.1254	0.2866	0.2060	0.1457	0.2013
ReFlow-DyRF	**21.63**	**26.29**	26.67	**20.77**	**27.95**	27.11	**25.07**
	0.8233	0.9043	0.9135	**0.7776**	**0.9330**	0.9192	**0.8785**
	0.2700	0.1850	0.1925	**0.3194**	0.1541	0.1669	**0.2146**
	0.2095	**0.1601**	0.1620	**0.1877**	**0.1320**	**0.1345**	**0.1643**

Table 8: Per-scene performance of various models with two input views on N3DV dataset. The four rows show PSNR, SSIM, LPIPS Alex, and Depth MAE scores, respectively.

Model \ Scene Name	Coffee Martini	Cook Spinach	Cut Roasted Beef	Flame Salmon	Flame Steak	Sear Steak	Average
K-Planes	16.20	17.56	22.32	13.36	18.88	19.36	17.95
	0.5295	0.5966	0.8413	0.3590	0.6899	0.7044	0.6201
	0.4553	0.4775	0.2253	0.6570	0.3967	0.3683	0.4300
	0.7790	0.5976	0.2024	0.9757	0.4730	0.4822	0.5850
K-Planes + Sparse Depth	15.87	17.34	20.94	12.65	17.65	17.66	17.02
	0.5396	0.6239	0.8078	0.3563	0.6310	0.6658	0.6041
	0.4477	0.4264	0.2547	0.6167	0.4151	0.3807	0.4235
	0.7115	0.4893	0.2193	**0.7521**	0.5576	0.4673	0.5328
DyRF	15.84	17.68	22.78	12.57	19.02	19.99	17.98
	0.5141	0.6070	0.8156	0.3481	0.6640	0.7030	0.6086
	0.4706	0.4591	0.2650	0.6424	0.4100	0.3849	0.4386
	0.8747	0.6107	0.3514	1.0873	0.5580	0.5492	0.6719
ReFlow-DyRF	**17.30**	**21.74**	**24.16**	**13.67**	**22.70**	**22.51**	**20.35**
	0.6005	**0.7707**	**0.8731**	**0.4097**	**0.8079**	**0.8190**	**0.7135**
	0.4165	**0.3046**	**0.1861**	**0.6103**	**0.2535**	**0.2381**	**0.3349**
	0.5743	**0.3404**	**0.1971**	0.7844	**0.3119**	**0.2759**	**0.4140**

Table 9: Per-scene performance of various models with four input views on N3DV dataset. The four rows show PSNR, SSIM, LPIPS Alex, and Depth MAE scores, respectively.

Model \ Scene Name	Coffee Martini	Cook Spinach	Cut Roasted Beef	Flame Salmon	Flame Steak	Sear Steak	Average
K-Planes	22.89	24.71	**28.64**	18.91	26.90	26.73	24.80
	0.8603	**0.8702**	**0.9418**	0.6985	**0.9040**	0.9051	0.8633
	0.1971	**0.1911**	**0.1068**	0.3657	**0.1632**	**0.1510**	0.1958
	0.3168	0.2588	0.1506	0.5428	**0.1578**	0.1730	0.2666
K-Planes + Sparse Depth	20.26	21.08	25.69	18.14	21.28	22.47	21.49
	0.7938	0.7696	0.9029	0.6907	0.7564	0.8052	0.7864
	0.2646	0.2957	0.1607	0.3608	0.2959	0.2454	0.2705
	0.2291	0.3044	**0.1493**	0.2761	0.3128	0.2662	0.2563
DyRF	22.02	23.36	27.91	17.70	25.72	25.86	23.76
	0.8468	0.8167	0.9295	0.6273	0.8679	0.8857	0.8290
	0.2290	0.2603	0.1306	0.4303	0.2069	0.1794	0.2394
	0.3545	0.3749	0.1590	0.6675	0.3359	0.2631	0.3591
ReFlow-DyRF	**23.43**	**25.76**	28.32	**22.85**	**27.10**	**27.68**	**25.86**
	0.8798	0.8699	0.9310	**0.8629**	0.8979	**0.9070**	**0.8914**
	0.2089	0.2101	0.1364	**0.2329**	0.1722	0.1621	**0.1871**
	0.1536	**0.2093**	0.1523	**0.1524**	0.2077	**0.1685**	**0.1740**

Table 10: Per-scene performance of various models with three input views on InterDigital dataset. The four rows show PSNR, SSIM, LPIPS Alex, and Depth MAE scores, respectively.

Model \ Scene Name	Birthday	Painter	Remy	Theater	Train	Average
K-Planes	19.58	26.42	15.18	17.01	15.56	18.75
	0.8639	0.9369	0.4302	0.5339	0.5591	0.6648
	0.1388	**0.0892**	0.5086	0.3949	0.3918	0.3047
	0.0886	**0.0700**	0.2936	0.2530	0.3846	0.2179
K-Planes + Sparse Depth	19.49	26.26	15.59	17.59	17.84	19.36
	0.8476	0.9305	0.4929	0.5744	0.6870	0.7065
	0.1525	0.0977	0.4580	0.3737	0.2792	0.2722
	0.1194	0.1708	0.2091	0.3036	0.2189	0.2044
DyRF	19.55	25.93	16.39	16.69	15.02	18.72
	0.8636	0.9272	0.5377	0.5248	0.5443	0.6795
	0.1533	0.1037	0.4413	0.4144	0.3990	0.3024
	0.0987	0.1344	0.2121	0.2608	0.3938	0.2200
DyRF + Sparse Depth	19.25	25.98	15.34	16.02	18.03	18.92
	0.8441	0.9249	0.5037	0.5338	0.6691	0.6951
	0.1615	0.1107	0.4566	0.4081	0.2908	0.2855
	0.1149	0.2323	0.2335	0.2832	0.2155	0.2159
DyRF + Cross-Cam Dense Flow	18.25	24.56	16.20	16.10	17.13	18.45
	0.7215	0.8613	0.5050	0.4545	0.6095	0.6304
	0.3034	0.1802	0.4735	0.4761	0.3677	0.3602
	0.2503	0.3576	0.3198	0.2397	0.2720	0.2879
ReFlow-DyRF w/o Sparse Flow	19.26	25.91	15.73	17.23	17.27	19.08
	0.8339	0.9099	0.4918	0.5346	0.6556	0.6851
	0.1946	0.1245	0.4974	0.4108	0.3138	0.3082
	0.1624	0.3070	0.2781	0.2833	0.2126	0.2487
ReFlow-DyRF w/o Dense Flow	19.51	26.51	**17.69**	18.45	18.14	20.06
	0.8723	**0.9397**	0.5594	0.5967	0.6901	0.7316
	0.1412	0.0917	0.4154	**0.3436**	0.2874	0.2559
	0.0666	0.0701	0.2036	0.2112	0.0775	0.1258
ReFlow-DyRF	**19.91**	**26.55**	17.51	**18.84**	**19.24**	**20.41**
	0.8680	0.9306	**0.5749**	**0.6092**	**0.7265**	**0.7418**
	0.1449	0.1038	**0.4036**	0.3471	**0.2546**	**0.2508**
	0.0718	0.0953	**0.1929**	**0.1952**	**0.0639**	**0.1238**

Table 11: Per-scene performance of various models with two input views on InterDigital dataset. The four rows show PSNR, SSIM, LPIPS Alex, and Depth MAE scores, respectively.

Model \ Scene Name	Birthday	Painter	Remy	Theater	Train	Average
K-Planes	16.71	20.23	14.18	16.24	15.20	16.51
	0.6798	0.6687	0.4016	0.4168	0.3991	0.5132
	0.2726	0.2905	0.5047	0.4573	0.4852	0.4021
	0.2105	0.3262	0.3790	0.3061	0.5471	0.3538
DyRF	17.76	21.27	15.00	16.12	15.06	17.04
	0.7572	0.7271	0.4778	0.4332	0.3958	0.5582
	0.2286	0.2487	**0.4659**	0.4600	0.4590	0.3725
	0.1782	0.3787	**0.3275**	0.3477	0.5409	0.3546
ReFlow-DyRF	**19.08**	**24.38**	**15.62**	**17.68**	**18.64**	**19.08**
	0.7913	**0.8360**	**0.4952**	**0.4905**	**0.6914**	**0.6609**
	0.2239	**0.1855**	0.4728	**0.4050**	**0.2837**	**0.3142**
	0.1688	**0.2542**	0.3404	**0.2870**	**0.1124**	**0.2326**

Table 12: Per-scene performance of various models with four input views on InterDigital dataset. The four rows show PSNR, SSIM, LPIPS Alex, and Depth MAE scores, respectively.

Model \ Scene Name	Birthday	Painter	Remy	Theater	Train	Average
K-Planes	23.26	**30.33**	21.04	**23.72**	20.93	**23.85**
	0.9294	**0.9708**	0.7512	**0.8288**	0.8163	**0.8593**
	0.0895	**0.0642**	0.2747	**0.1897**	0.1674	**0.1571**
	0.0553	**0.0547**	0.1549	**0.1475**	0.1448	0.1114
DyRF	**23.31**	29.60	21.46	22.45	20.72	23.51
	0.9243	0.9662	**0.7649**	0.7598	0.8114	0.8453
	0.1013	0.0725	**0.2552**	0.2444	0.1879	0.1723
	0.0697	0.0965	0.1431	0.1980	0.1385	0.1292
ReFlow-DyRF	23.02	29.22	**21.74**	22.82	**21.65**	23.69
	0.9228	0.9671	0.7571	0.7748	**0.8546**	0.8553
	0.0992	0.0688	0.2606	0.2484	**0.1541**	0.1662
	0.0519	0.0706	**0.1398**	0.1776	**0.0632**	**0.1006**