KFD-NeRF: Rethinking Dynamic NeRF with Kalman Filter Supplemental Material

Yifan Zhan[1], Zhuoxiao Li[1], Muyao Niu[1], Zhihang Zhong[3], Shohei Nobuhara[2], Ko Nishino[2], and Yinqiang Zheng[⋆1]

[1] The University of Tokyo
[2] Kyoto University
[3] Shanghai Artificial Intelligence Laboratory

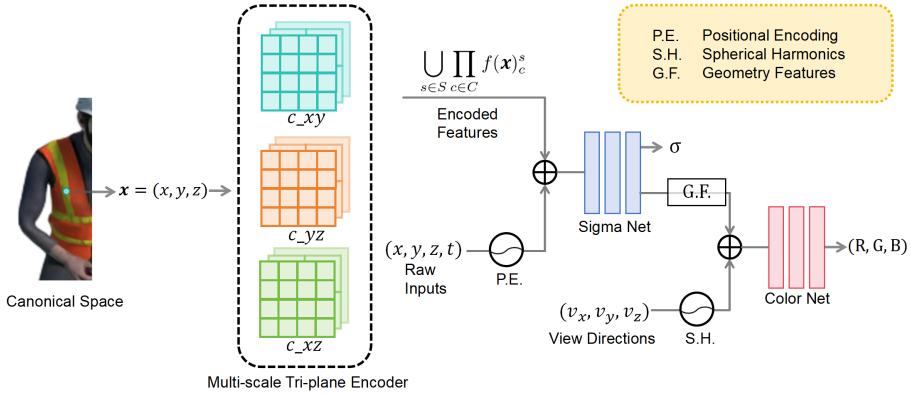

Fig. 1: Details of our spatial neural architectures.

1 Per-scene Results

We exhibit our per-scene results for synthetic data in Tab. 1 and for real data in Tab. 2. In Fig. 2 and Fig. 3 we additionally provide more visualization results on synthetic and real data.

2 Spatial Neural Architectures

We elaborate in detail on our spatial neural architectures in Fig. 1. Once obtaining the warped 3D points in the canonical space, we first use a multi-scale tri-plane to encode the spatial information. For each plane $c \in C$ and each scale $s \in S$, features are multiplied across planes and concatenated across scales to obtain tri-plane encoded features. However, coordinate shifts happen due to the limited resolution of the tri-plane grids and errors introduced by linear interpolation. We follow [1] by concatenating encoded tri-plane features with positional

	Bouncing Balls				Hell Warrior				Hook				Jumping Jacks			
Method	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓
D-NeRF [6]	38.05	0.982	0.107	0.018	24.94	0.948	0.071	0.038	29.43	0.962	0.122	0.032	31.96	0.972	0.045	0.019
TiNeuVox-B [1]	40.53	0.990	0.036	0.007	28.21	0.965	0.072	0.027	32.31	0.975	0.044	0.016	34.70	0.983	0.033	0.012
KPlanes [2]	41.12	0.991	0.032	0.006	25.65	0.952	0.079	0.036	28.55	0.957	0.082	0.029	32.60	0.977	0.054	0.017
NDVG [4]	34.59	0.969	0.113	0.019	25.58	0.949	0.075	0.037	29.74	0.966	0.040	0.021	29.55	0.960	0.081	0.027
V4D [3]	42.00	**0.992**	**0.029**	**0.005**	27.06	0.960	0.054	0.028	30.95	0.972	0.037	0.018	35.29	0.986	**0.022**	**0.010**
4D-GS [9]	39.18	0.990	0.033	0.007	27.53	0.968	**0.042**	0.024	32.13	0.978	**0.022**	0.013	34.71	0.986	**0.022**	**0.010**
KFD-NeRF	**42.16**	0.991	0.038	0.006	**30.58**	**0.978**	0.051	**0.019**	**35.65**	**0.990**	0.034	**0.010**	**35.64**	**0.988**	0.041	0.011
	Lego				Mutant				Stand Up				T-Rex			
Method	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓
D-NeRF [6]	21.70	0.842	0.169	0.077	31.19	0.973	0.029	0.016	33.41	0.980	0.023	0.012	31.30	0.972	0.043	0.018
TiNeuVox-B [1]	25.17	0.924	0.075	0.040	33.76	0.979	0.031	0.013	36.03	0.985	0.021	0.009	32.74	0.979	0.033	0.014
KPlanes [2]	25.52	**0.948**	0.059	0.034	24.71	0.917	0.178	0.057	33.89	0.983	0.051	0.014	31.71	0.981	0.038	0.016
NDVG [4]	25.21	0.933	0.052	0.034	35.44	0.988	0.015	0.008	34.01	0.982	0.023	0.011	30.00	0.967	0.048	0.021
V4D [3]	**25.63**	**0.948**	**0.038**	**0.029**	36.14	0.989	0.014	0.007	37.06	0.990	0.012	**0.006**	34.21	0.987	**0.018**	**0.010**
4D-GS [9]	25.37	0.940	0.044	0.031	37.80	0.993	**0.009**	**0.005**	36.82	0.990	**0.011**	0.007	32.97	0.984	0.022	0.012
KFD-NeRF	25.54	0.948	0.070	0.035	**39.23**	**0.995**	0.039	0.007	**39.62**	**0.994**	0.030	0.007	**37.40**	**0.992**	0.054	**0.010**

Table 1: Per-scene quantitative comparisons on synthetic dynamic scenes.

	Balloon1				Balloon2				dynamicFace				Jumping			
Method	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓
TiNeuVox-B [1]	25.21	0.773	0.249	0.071	26.35	0.814	0.208	0.059	22.30	0.887	0.167	0.069	25.41	0.779	0.329	0.077
KPlanes [2]	28.20	0.887	0.100	0.037	26.85	0.863	0.157	0.044	25.44	0.923	0.112	0.045	27.09	0.857	0.206	0.058
Mixvoxels-L [8]	26.24	0.808	0.235	0.063	26.78	0.811	0.235	0.060	20.03	0.792	0.308	0.112	26.91	0.855	0.230	0.059
V4D [3]	27.11	0.888	0.101	0.040	24.55	0.847	0.148	0.059	**27.20**	**0.951**	**0.083**	**0.033**	**27.78**	**0.883**	**0.175**	**0.049**
KFD-NeRF	**28.83**	**0.906**	**0.076**	**0.031**	**27.30**	**0.888**	**0.089**	**0.038**	26.45	0.937	0.092	0.037	26.93	0.845	0.215	0.061
	Playground				Skating				Truck				Umbrella			
Method	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓	PSNR↑	SSIM↑	LPIPS↓	Average↓
TiNeuVox-B [1]	16.60	0.376	0.461	0.200	27.93	0.840	0.276	0.056	25.78	0.765	0.356	0.077	25.56	0.636	0.381	0.086
KPlanes [2]	24.59	0.836	0.152	0.060	34.06	0.956	0.092	0.020	32.93	0.925	0.118	0.025	26.84	0.801	0.159	0.057
Mixvoxels-L [8]	23.19	0.748	0.253	0.085	33.14	0.945	0.148	0.026	32.60	0.920	0.133	0.027	26.70	0.748	0.236	0.063
V4D [3]	**25.69**	**0.875**	**0.100**	**0.046**	33.79	0.956	0.115	0.022	32.95	0.930	0.126	0.026	26.24	0.742	0.237	0.066
KFD-NeRF	24.59	0.846	0.129	0.056	**34.95**	**0.964**	**0.077**	**0.017**	**33.44**	**0.930**	**0.100**	**0.023**	**27.49**	**0.815**	**0.146**	**0.048**

Table 2: Per-scene quantitative comparisons on real dynamic scenes.

Model	PSNR(dB)↑	SSIM↑	LPIPS↓	Training Time↓	Params(MB)↓
MLP Based	32.13	0.971	0.052	43hrs	**0.5**
Voxel Grid Based	36.21	0.984	**0.029**	**60mins**	2336
Tri-plane Based	**38.92**	**0.990**	0.037	150mins	31.5

Table 3: Ablation Study on Spatial Representations. We compare three different spatial models on rendering quality, training time and spatial-only size. The tri-plane model we choose significantly reduces the model size compared to the voxel grid based model and achieves the highest rendering quality in a relatively short training time.

encoded [7] raw inputs. Our Sigma Net (single-hidden-layer MLP) outputs volume density σ and 15-dimensional geometry features. The geometry features will be further concatenated with spherical hamonics encoded view directions for color calculation using Color Net (two-hidden-layer MLP).

3 Model Hyperparameters

The frequency number of positional encoding is set to 5 for both spatial and temporal inputs. Our shallow observation MLP consists of two hidden layers,

each has a channel dimension of 128. As for the tri-plane grids, we use multi-scale planes with 4 different resolutions at 64^2, 128^2, 256^2 and 512^2. As can be seen in Fig. 1, the per-plane and per-scale features are multiplied across planes and concatenated across scales. We set each of these per-plane and per-scale features' dimension to be 32 so the final encoded feature has dimension 128. Sigma Net has single hidden layer with channel dimension 64 and Color Net has two hidden layers with channel dimension 64.

For optimization, an Adam optimizer [5] is used and we set ray batch to be 4096 in each iteration. We train our KFD-NeRF with learning rate set to 1×10^{-3}.

4 Ablation Study on Spatial Representations

In Section 4.3, we analyze the characteristics of different spatial representations and their reconstruction capabilities in the canonical space. Based on our full model, we further conduct ablation studies by only changing spatial representations to show their impacts on reconstruction results.

Specifically, we compare three different spatial models, namely, pure MLP (8 hidden layers with dimension 256), multi-scale voxel grid (resolutions at 64^3, 128^3 and 256^3) and multi-scale tri-plane (resolutions at 64^2, 128^2 and 256^2). In Tab. 3, We compare these models based on three dimensions: model size, convergence time, and rendering quality.

Our full model only uses a shallow MLP with two hidden layers to calculate deformations, so the pure spatial MLP would suffer from convergence difficulties. The voxel grid based model converges faster while the model size is far from acceptable. Therefore we choose the tri-plane based model as a spatial model that converges relatively fast, has a moderate size, and provides high rendering quality.

References

1. Fang, J., Yi, T., Wang, X., Xie, L., Zhang, X., Liu, W., Nießner, M., Tian, Q.: Fast Dynamic Radiance Fields with Time-aware Neural Voxels. In: SIGGRAPH Asia 2022 Conference Papers. pp. 1–9 (2022)
2. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit Radiance Fields in Space, Time, and Appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12479–12488 (2023)
3. Gan, W., Xu, H., Huang, Y., Chen, S., Yokoya, N.: V4D: Voxel for 4D Novel View Synthesis. IEEE Transactions on Visualization and Computer Graphics (2023)
4. Guo, X., Chen, G., Dai, Y., Ye, X., Sun, J., Tan, X., Ding, E.: Neural Deformable Voxel Grid for Fast Optimization of Dynamic View Synthesis. In: Proceedings of the Asian Conference on Computer Vision. pp. 3757–3775 (2022)
5. Kingma, D.P., Ba, J.: Adam: A method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)

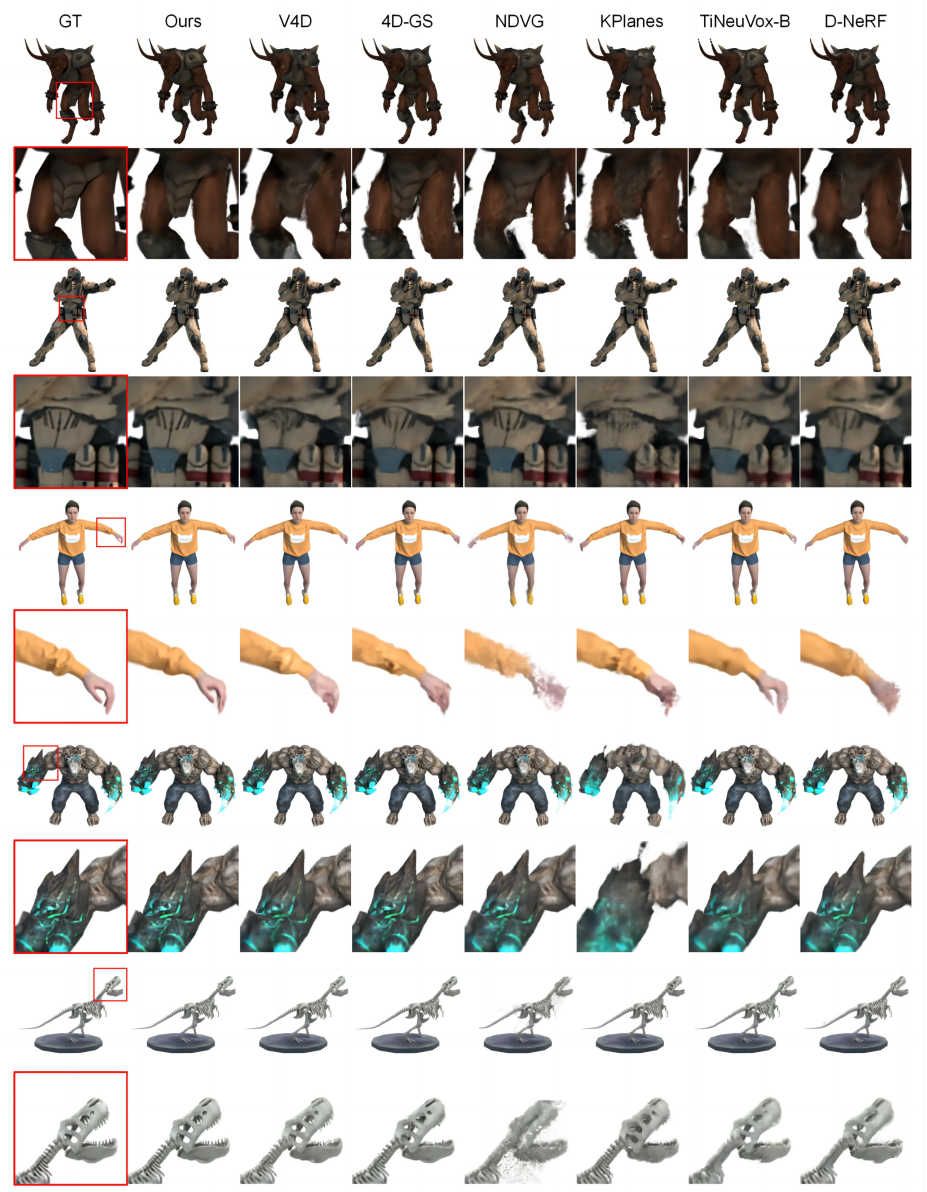

Fig. 2: More qualitative results on synthetic data. Zoom in for better details.

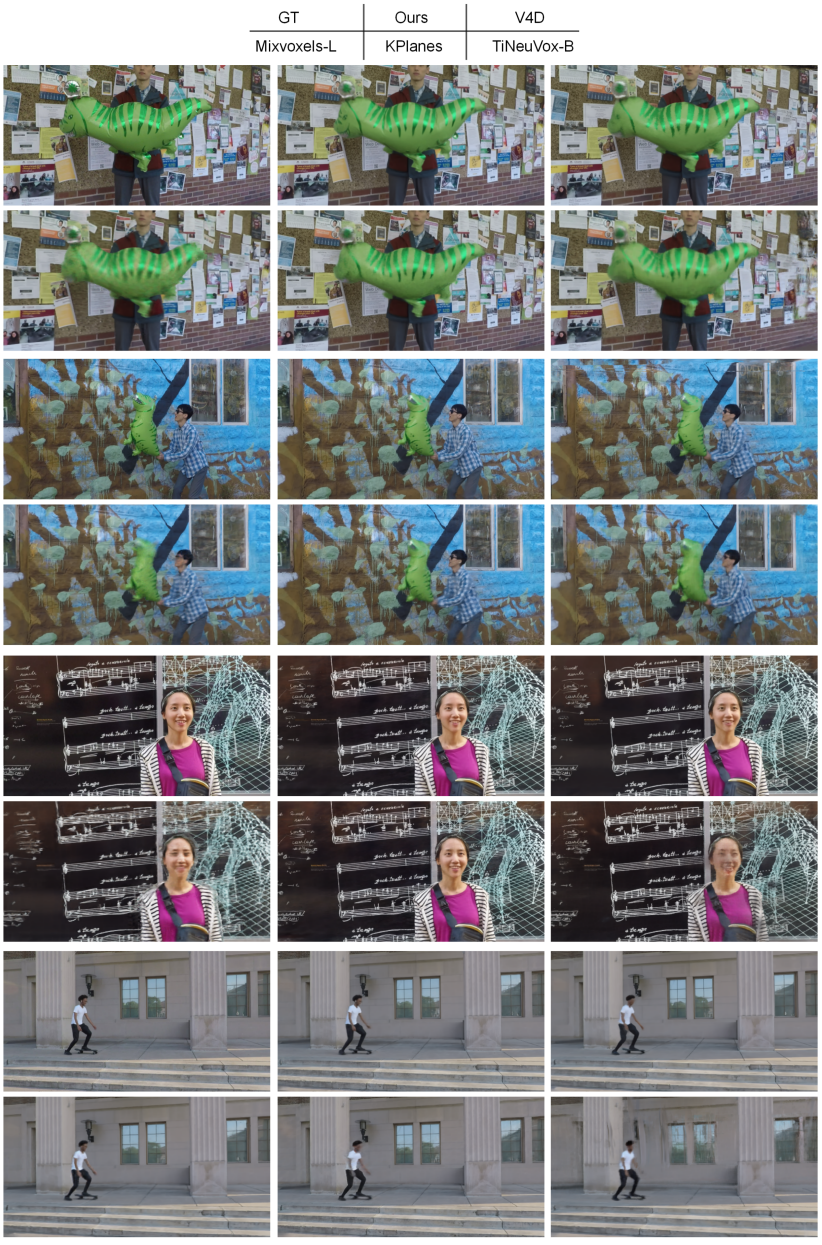

Fig. 3: More qualitative results on real data. Zoom in for better details.

6. 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)
7. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the Spectral Bias of Neural Networks. In: International Conference on Machine Learning. pp. 5301–5310. PMLR (2019)
8. Wang, F., Tan, S., Li, X., Tian, Z., Song, Y., Liu, H.: Mixed Neural Voxels for Fast Multi-view Video Synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19706–19716 (2023)
9. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. arXiv preprint arXiv:2310.08528 (2023)