Supplementary Materials: 4D Gaussian Splatting with Scale-aware Residual Field and Adaptive Optimization for Real-time rendering of temporally complex dynamic scenes

Anonymous Author(s)
Submission Id: 4096

1 OVERVIEW

With in the supplemtantary, we provide:

- Details of Adaptive Optimization in Sec. 2
- Hyperparameter Settings in Sec. 3
- More Results in Sec. 4

2 DETAILS OF ADAPTIVE OPTIMIZATION

Based on the unique temporal characteristics of each Gaussian primitive, we apply distinct optimization schedules. Integrating the state function over time intervals enables us to represent the sampling probability of Gaussian primitives in the temporal domain: $I = F(t_{end}) - F(t_{start})$, Here, $F(t)$ represents the cumulative distribution function (CDF) of the Gaussian primitive's state function $\gamma(t)$.

$$F(t) = P(x < t) = \int_{-\infty}^{t} e^{-k\frac{x-\tau}{\sigma}^2} dx, \tag{1}$$

We approximate $F(t)$ based on[10]:

$$Q(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = 1 - \frac{1}{e^{1+\alpha_1 t^3 + \alpha_2 t}}, \tag{2}$$

To transform it into the form of $F(t)$, we employ the method of integration by substitution:

$$x = \sqrt{2k}\frac{m-\tau}{\sigma} \tag{3}$$

$$\begin{aligned}
Q(t) &= \int_{-\infty}^{\frac{\sigma t}{\sqrt{2k}}+\tau} \frac{1}{\sqrt{2\pi}} e^{-\frac{(\sqrt{2k}\frac{m-\tau}{\sigma})^2}{2}} d(\sqrt{2k}\frac{m-\tau}{\sigma}) && (4)\\
&= \int_{-\infty}^{\frac{\sigma t}{\sqrt{2k}}+\tau} \frac{1}{\sqrt{2\pi}} e^{-k\frac{m-\tau}{\sigma}^2} d(\sqrt{2k}\frac{m-\tau}{\sigma}) && (5)\\
&= \int_{-\infty}^{\frac{\sigma t}{\sqrt{2k}}+\tau} \frac{1}{\sqrt{2\pi}} e^{-k\frac{m-\tau}{\sigma}^2} d(\sqrt{2k}\frac{m-\tau}{\sigma}) && (6)\\
&= \int_{-\infty}^{\frac{\sigma t}{\sqrt{2k}}+\tau} \frac{1}{\sqrt{2\pi}} \frac{\sqrt{2k}}{\sigma} e^{-k\frac{m-\tau}{\sigma}^2} dm && (7)\\
&= \frac{\sqrt{k}}{\sqrt{\pi}\sigma} \int_{-\infty}^{\frac{\sigma t}{\sqrt{2k}}+\tau} e^{-k\frac{m-\tau}{\sigma}^2} dm && (8)\\
&= \frac{\sqrt{k}}{\sqrt{\pi}\sigma} F(\frac{\sigma t}{\sqrt{2k}} + \tau), && (9)
\end{aligned}$$

Therefore, we can obtain $F(t)$:

$$F(t) = \frac{\sqrt{\pi}\sigma}{\sqrt{k}} Q(\sqrt{2k}\frac{(t-\tau)}{\sigma}). \tag{10}$$

For each Gaussian primitive $\mathcal{G}_i^{4D}$ with distinct σ_i and τ_i, we can derive $F_i(t)$ based on Eq. 10. Then, we obtain I_i for each $\mathcal{G}_i^{4D}$, thus adopting different optimization schedules for each Gaussian primitive.

3 HYPERPARAMETERS SETTINGS

We predominantly adhere to the hyperparameter settings of 3DGS[6]. Specifically, the batch size in training is set to 4 , and we initialize the learning rate of Scale-aware Residual Field parameters at 3.2e-3, which decays to 3.2e-6 by the end of training. Similarly, we initialize the learning rates of all tiny MLP decoders at 1.6e-4, decaying to 1.6e-7 by the end of training. Furthermore, we opt to abandon the strategy of filtering out larger Gaussians in worldspace. Our decision stems from the observation that this strategy results in incomplete backgrounds in our framework, thereby compromising our rendering quality.

Different datasets are collected under different settings, corresponding to different initialization methods. For the D-NeRF dataset[11], which involves monocular synthesized scene data, we uniformly and randomly initialize 10,000 Gaussian primitives distributed within a cube of $[-1.3, 1.3]^3$. Additionally, the temporal position of each Gaussian is uniformly initialized within the range of 0 to 1. We adopt a warm-up strategy with 1,000 iterations to train the scene as static, compensating for geometric information loss due to the absence of initialized point clouds. We trained these synthesized scenes using a black background. Additionally, we set the opacity reset interval to 2,000 iterations to accelerate training. The total training duration is set to 20,000 iterations.

For the multi-view real-world Plenoptic Video dataset[7], we utilize point clouds generated by COLMAP as our initialization point cloud. To achieve more accurate scene boundaries, in addition to using point clouds from the first frame, we incorporate sparse point clouds generated from subsequent frames after undergoing sparse filtering. The total number of initial points for each scene is around 40,000. The temporal position of each Gaussian is also uniformly initialized within the range of 0 to 1. In this setting, warm-up is not necessary. We set the opacity reset interval to 3,000 iterations to align with 3DGS[6].

4 MORE RESULTS

The evaluation results for D-NeRF are illustrated in Fig. 1. This figure contains a typo in the full paper, and we have corrected it here. Comparing with state-of-the-art (SOTA) methods[2–5, 11, 14], the per-scene evaluation results are presented in Tab. 1 for the D-NeRF dataset. Similarly, for the Plenoptic Video dataset, the per-scene evaluation results compared with SOTA methods[1, 2, 4, 8, 9, 12–15] are shown in Tab. 2.

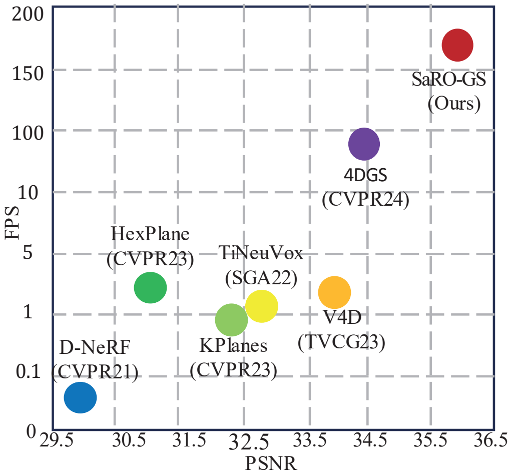

Figure 1: Comparison on Quality and Speed. There is a typo in the full paper, and we have corrected it here.

In comparison with [14], more qualitative comparisons are presented in Fig. 2. More results regarding dynamic-static segmentation in real-world scenes are presented in Fig. 3. Additionally, the depth map can alse be obtained through Gaussian splatting during the rendering process,as shown in Fig. 3.

We rendered a video of dynamic scenes using the flame steak scene of Plenoptic Video dataset as an example. The viewpoints were uniformly sampled on a sphere to validate the capabilities of our SaRO-GS in free-viewpoint interaction with dynamic scenes. The results are shown in the video file 'SaRO_Free_trajectory_example.mp4'.

REFERENCES

[1] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. 2023. HyperReel: High-fidelity 6-DoF video with ray-conditioned sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16610–16620.

[2] 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*. 130–141.

[3] 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 *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

[4] 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*. 12479–12488.

[5] Wanshui Gan, Hongbin Xu, Yi Huang, Shifeng Chen, and Naoto Yokoya. 2023. V4d: Voxel for 4d novel view synthesis. *IEEE Transactions on Visualization and Computer Graphics* (2023).

[6] 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), 1–14.

[7] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Richard Newcombe. 2022. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5521–5531.

[8] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. 2023. Spacetime Gaussian Feature Splatting for Real-Time Dynamic View Synthesis. , arXiv:2312.16812 pages. https://doi.org/10.48550/arXiv.2312.16812 Project page: https://oppo-us-research.github.io/SpacetimeGaussians- website/.

[9] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. 2023. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713* (2023).

[10] E. Page. 2018. Approximations to the Cumulative Normal Function and its Inverse for Use on a Pocket Calculator. *Journal of the Royal Statistical Society Series C: Applied Statistics* 26, 1 (2018), 75–76. https://doi.org/10.2307/2346872

[11] 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*. 10318–10327.

[12] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. 2023. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics* 29, 5 (2023), 2732–2742.

[13] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. 2023. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19706–19716.

[14] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 2023. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528* (2023).

[15] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. 2023. Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting. , arXiv:2310.10642 pages. https://doi.org/10.48550/arXiv.2310.10642 Technical Report.

Table 1: The per-scene evaluation results on the D-NeRF dataset.† denotes a dynamic Gaussian method.

Method	Bouncing Balls			Hellwarrior			Hook			Jumpingjacks		
	PSNR(dB)↑	SSIM↑	LPIPS↓	PSNR(dB)↑	SSIM↑	LPIPS↓	PSNR(dB)↑	SSIM↑	LPIPS↓	PSNR(dB)↑	SSIM↑	LPIPS↓
D-NeRF[11]	38.93	0.98	0.10	25.02	0.95	0.06	29.25	0.96	0.11	32.80	0.98	0.03
KPlanes-hybrid[4]	40.33	0.99	-	24.81	0.95	-	28.13	0.95	-	31.64	0.97	-
TiNeuVox-B[3]	40.73	0.99	0.04	28.17	0.97	0.07	31.45	0.97	0.05	34.23	0.98	0.03
V4D[5]	42.67	0.99	0.02	27.03	0.96	0.05	31.04	0.97	0.03	35.36	0.99	0.02
HexPlane[2]	39.69	0.99	0.03	24.24	0.94	0.07	28.71	0.96	0.05	31.65	0.97	0.04
4DGS[14]†	40.62	0.99	0.02	28.71	0.97	0.04	32.73	0.98	0.03	35.42	0.99	0.01
Ours	36.02	0.99	0.01	38.01	0.97	0.02	36.81	0.99	0.01	34.56	0.98	0.016
Method	Lego			Mutant			Standup			Trex		
	PSNR(dB)↑	SSIM↑	LPIPS↓	PSNR(dB)↑	SSIM↑	LPIPS↓	PSNR(dB)↑	SSIM↑	LPIPS↓	PSNR(dB)↑	SSIM↑	LPIPS↓
D-NeRF[11]	21.64	0.83	0.16	31.29	0.97	0.02	32.79	0.98	0.02	31.75	0.97	0.03
KPlanes-hybrid[4]	25.27	0.94	-	32.59	0.97	-	33.17	0.98	-	30.75	0.97	-
TiNeuVox-B[3]	25.02	0.92	0.07	33.61	0.98	0.03	35.43	0.99	0.02	32.70	0.98	0.03
V4D[5]	25.62	0.95	0.04	36.27	0.99	0.01	37.20	0.99	0.01	34.53	0.99	0.02
HexPlane[2]	25.22	0.94	0.04	33.79	0.98	0.03	34.36	0.98	0.02	30.67	0.98	0.03
4DGS[14]†	25.03	0.94	0.04	37.59	0.99	0.02	38.11	0.99	0.01	34.23	0.99	0.01
Ours	25.46	0.94	0.04	42.11	0.99	0.01	44.45	0.99	0.01	31.62	0.98	0.01

Table 2: The per-scene evaluation results on the Plenoptic Video dataset.† denotes a dynamic Gaussian method.

Method	Coffee Martini			Cook Spinach			Cut Beef		
	PSNR(dB)↑	DSSIM↓	LPIPS↓	PSNR(dB)↑	DSSIM↓	LPIPS↓	PSNR(dB)↑	DSSIM↓	LPIPS↓
KPlanes-hybrid[4]	29.99	0.024	-	32.60	0.017	-	31.82	0.017	-
Mix-Voxels-L[13]	29.63	0.024	0.106	32.25	0.016	0.099	32.40	0.016	0.088
NeRFPlayer[12]	31.53	-	0.085	30.56	-	0.113	29.35	-	0.144
HyperReel[1]	28.37	-	0.127	32.30	-	0.089	32.92	-	0.084
HexPlane[2]	-	-	-	32.04	0.015	0.082	32.55	0.013	0.080
Dynamic 3DGS[9]†	26.49	0.033	0.139	32.97	0.013	0.087	30.72	0.016	0.090
4DGS-Realtime[15]†	28.33	-	-	32.93	-	-	33.85	-	-
Spacetime-Gs[8]†	28.61	0.025	0.069	33.18	0.011	0.037	33.52	0.011	0.036
4DGS[14]†	27.34	0.048	-	32.46	0.026	-	32.90	0.022	-
Ours	28.96	0.021	0.061	33.19	0.012	0.038	33.91	0.021	0.038
Method	Flame Salmon			Flame Steak			Sear Steak		
	PSNR(dB)↑	DSSIM↓	LPIPS↓	PSNR(dB)↑	DSSIM↓	LPIPS↓	PSNR(dB)↑	DSSIM↓	LPIPS↓
KPlanes-hybrid[4]	30.44	0.024	-	32.38	0.015	-	32.52	0.013	-
Mix-Voxels-L[13]	29.81	0.026	0.116	31.83	0.014	0.088	32.10	0.012	0.080
NeRFPlayer[12]	31.65	-	0.098	31.93	-	0.088	29.13	-	0.138
HyperReel[1]	25.02	-	0.136	33.61	-	0.078	35.43	-	0.077
HexPlane[2]	29.47	0.018	0.078	31.82	0.012	0.071	32.23	0.012	0.070
Dynamic 3DGS[9]†	26.92	0.030	0.122	33.24	0.011	0.079	33.68	0.011	0.079
4DGS-Realtime[15]†	29.38	-	-	34.03	-	-	33.51	-	-
Spacetime-Gs[8]†	29.48	0.022	0.063	33.64	0.009	0.029	33.89	0.009	0.030
4DGS[14]†	29.20	0.042	-	32.51	0.023	-	32.49	0.002	-
Ours	29.14	0.021	0.059	33.83	0.010	0.034	33.89	0.010	0.036

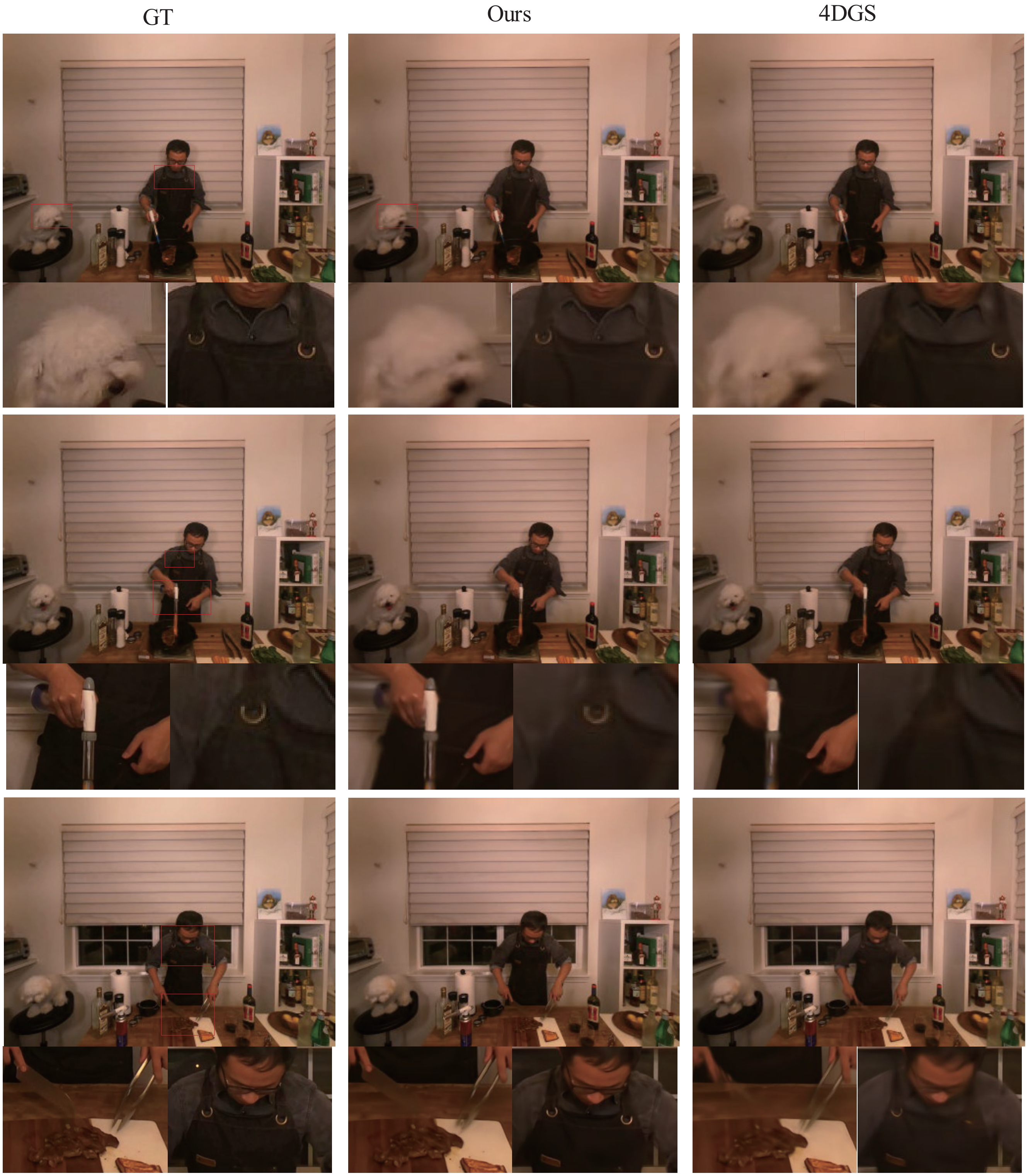

Figure 2: Qualitative comparison with [14] on the Plenoptic Video dataset.

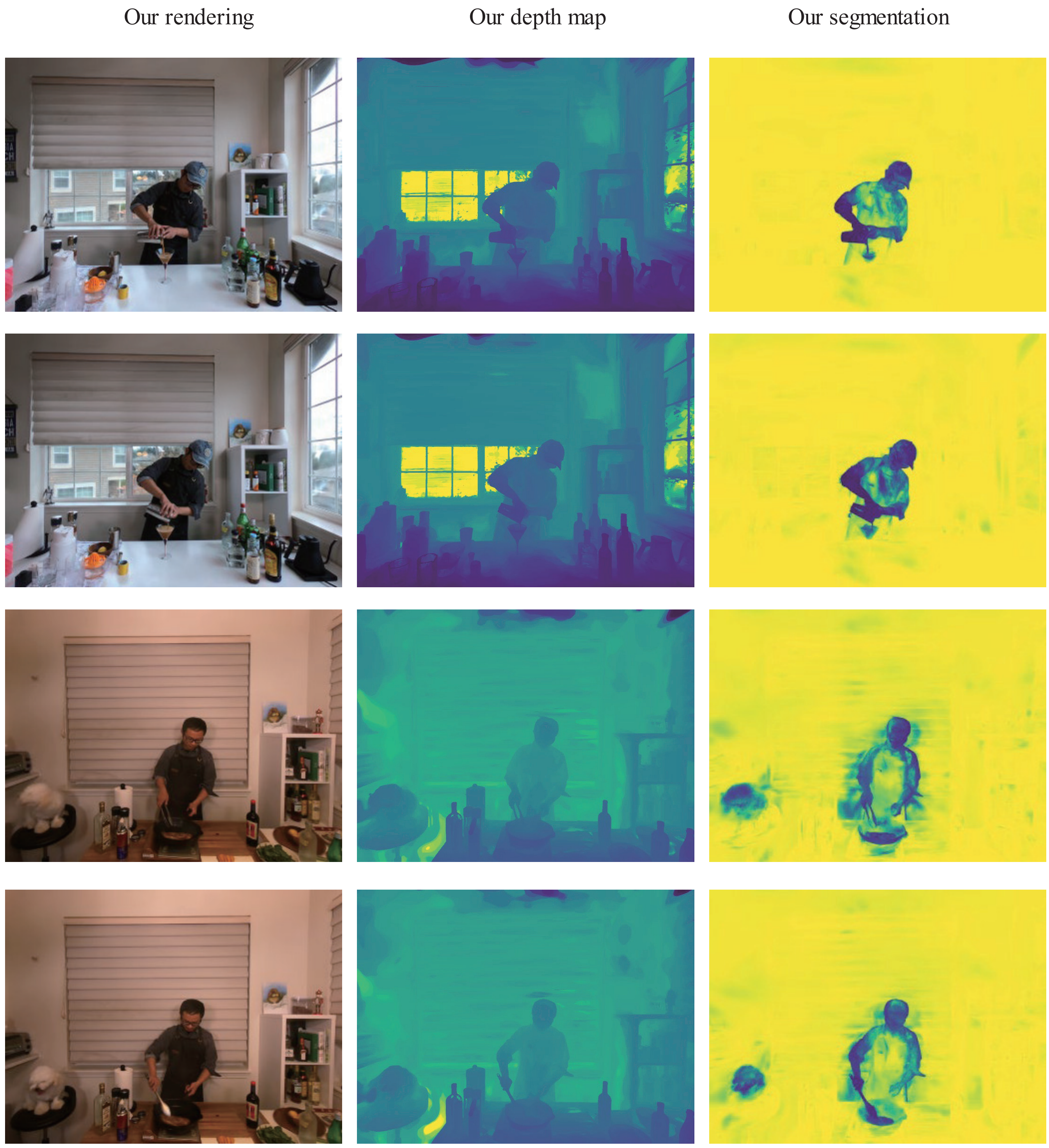

Figure 3: Depth map and Segmentation of dynamic and static scenes.