SplatFlow: Self-Supervised Dynamic Gaussian Splatting in Neural Motion Flow Field for Autonomous Driving

Supplementary Material

In this supplementary material, we provide more implementation details of SplatFlow in Appendix 1. We provide more visualization of 3D LiDAR points within NMFF in Appendix 2. We present more detailed comparison visualizations of dynamic object synthesis from novel views in Appendix 3. We provide more visualizations of dynamic objects decomposition in Appendix 4. We present visualizations of the novel view synthesis on newly generated ego-car trajectories in Appendix 5. We include visualizations of rendered RGB image, optical flow and depth in Appendix 6. Runtime performance comparisons are provided in Appendix 7. Video demonstrations are included in Appendix 8.

1. Implementation Details

Each field in the Neural Motion Flow Field (NMFF) consists of eight ReLU-MLP stacks. All MLPs are followed by a ReLU activation, except for the final prediction layer, where the middle hidden dimensions are configured as 128.

For NMFF pre-training, we follow the approach in [5] to generate pseudo scene flow labels, excluding ground points from the Waymo and KITTI datasets. The raw 3D LiDAR points are utilized without cropping to a smaller range. We use a learning rate of 8e-3 with the Adam optimizer, optimizing each scene for up to 4000 iterations with early stopping. Additionally, point cloud densification is performed by accumulating point clouds through Euler integration, using per-pair scene flow estimations.

During the 4D Gaussian with NMFF optimization, we configure the position learning rate to a range from 1.6e-5 to 1.6e-6, the opacity learning rate to 0.05, the scale learning rate to 0.005, the feature learning rate to 2.5e-3, and the rotation learning rate to 0.001. The intervals for densification and opacity reset are set to 500 and 3000, respectively. We set the densify gradient threshold for decomposed static and dynamic Gaussians as 1.7e-4 and 1e-4, respectively. The Spherical Harmonics degree for each Gaussian is set to 3. For NMFF optimization, we set the learning rate to 1e-4. For training losses, we use coefficients $\lambda_1 = 0.1$, $\lambda_2 = 0.005$, $\lambda_3 = 0.05$, $\lambda_4 = 0.001$, $\lambda_{ssim} = 0.2$ and $\lambda_f = 0.8$.

2. Visualization of LiDAR Points in NMFF

We provide more visualization of 3D LiDAR points within NMFF on the Waymo dataset in Fig. 1. The color wheel in the center represents the flow magnitude through color intensity and the flow direction via angle. As illustrated, NMFF accurately predicts the 3D motion flow of 3D LiDAR points for dynamic objects in driving scenarios.

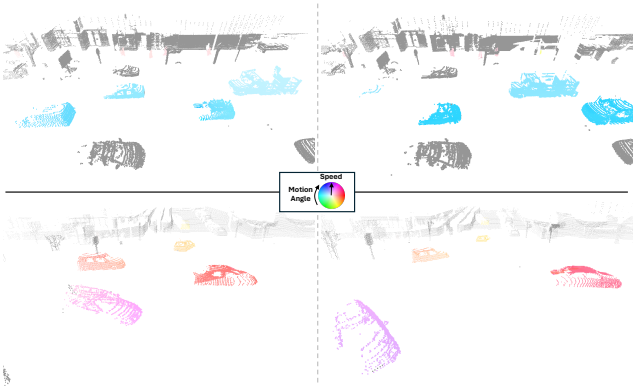


Figure 1. Visualization of 3D LiDAR points within NMFF on Waymo dataset.

Figure 2. Dynamic object decomposition results of SplatFlow on Waymo. Row1: Rendered scene, Row2: Corresp. Decomposition

	Waymo FPS	KITTI FPS
S-NeRF [8]	0.0014	0.0075
StreetSurf [2]	0.097	0.37
NSG [4]	0.032	0.19
Mars [7]	0.030	0.31
SUDS [6]	0.008	0.04
EmerNerf [9]	0.053	0.28
3DGS [3]	**63**	**125**
PVG [1]	50	59
SplatFlow	40	44

Table 1. The comparison running-time analysis on Waymo and KITTI datasets.

Figure 3. Novel view synthesis on newly generated ego-car trajectories on Waymo dataset.

3. Visual Comparison of Dynamic Object Synthesis

We present more visual comparison details of dynamic object synthesis from novel views, showcasing results on the Waymo dataset in Fig. 4 and on the KITTI dataset in Fig. 5. The dynamic objects in these figures are selected from a distant background car in an extremely zoomed-in view. As shown, SplatFlow generates sharper images with fewer blurred artifacts, particularly for high-speed vehicles, compared to the baselines.

4. Visualization of Dynamic Object Decomposition

We provide more dynamic object decomposition visualization of SlatFlow in Fig.2. As shown, SplatFlow clearly separates the dynamic objects from the rendered scenes.

5. Visualization of Novel View Synthesis on newly Ego-car Trajectory

We provide novel view synthesis in a more challenging rendering scenario on the Waymo dataset, using newly generated ego-car trajectories. As shown in Fig. 3, the results are produced from an ego-car trajectory that is shifted 0.5 meters higher, 0.5 meters lower, 0.5 meters to the left, and 0.5 meters to the right of the original trajectory. As demonstrated, SplatFlow can render high-quality novel images for these newly generated ego-car trajectories.

6. Rendered Depth and Flow Visualization

We also present visualization of rendered RGB image, optical flow, and depth on the Waymo dataset in Fig. 6 and 7, and on the KITTI dataset in Fig. 8. In these Figures, the first row displays the rendered RGB images, the second row shows the rendered optical flow, and the third row presents the rendered depth, all generated by SplatFlow. As shown, our SplatFlow renders sharp, clear, and dense optical flow and depth images in dynamic driving scenarios.

7. Running-time Analysis

We compare the runtime performance of SplatFlow with various baseline methods on the Waymo and KITTI datasets, as summarized in Table 1. Utilizing a single NVIDIA GeForce A6000, SplatFlow achieves real-time rendering speeds for high-resolution images after quantization and pruning optimization, delivering approximately 40 FPS at 1920×1280 resolution on the Waymo dataset and around 44 FPS at 1242×375 resolution on the KITTI dataset. Compared to NeRF-based methods such as S-NeRF [8], StreetSurf [2], NSG [4], SUDS [6], EmerNerf [9], SplatFlow significantly surpasses the speed of these methods, delivering real-time performance. Compared to GS-based methods such as 3DGS [3] and PVG [1], SplatFlow achieves higher accuracy in dynamic object rendering

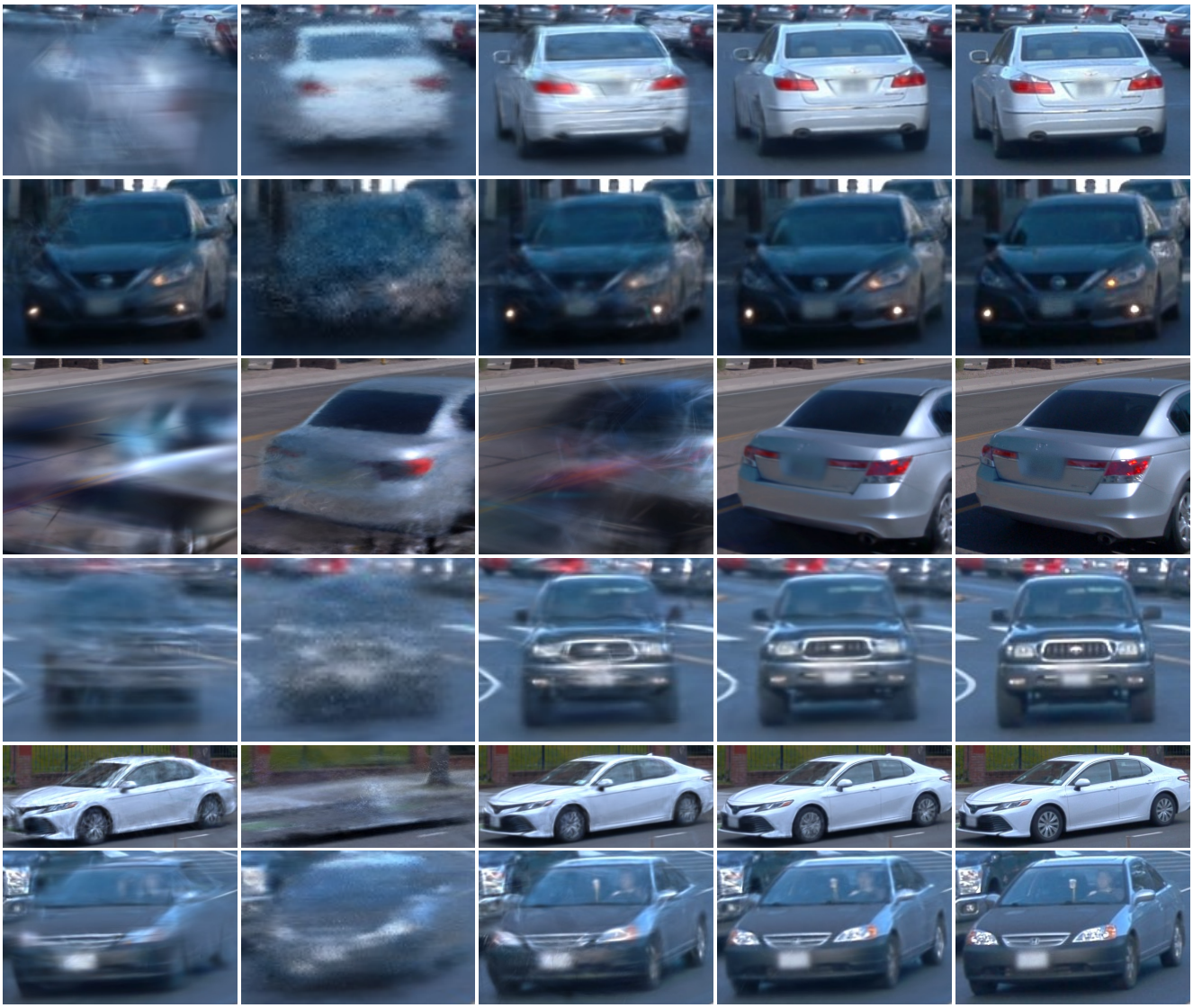


Figure 4. Detailed comparison of dynamic object synthesis from novel views on Waymo dataset.

while maintaining efficient performance.

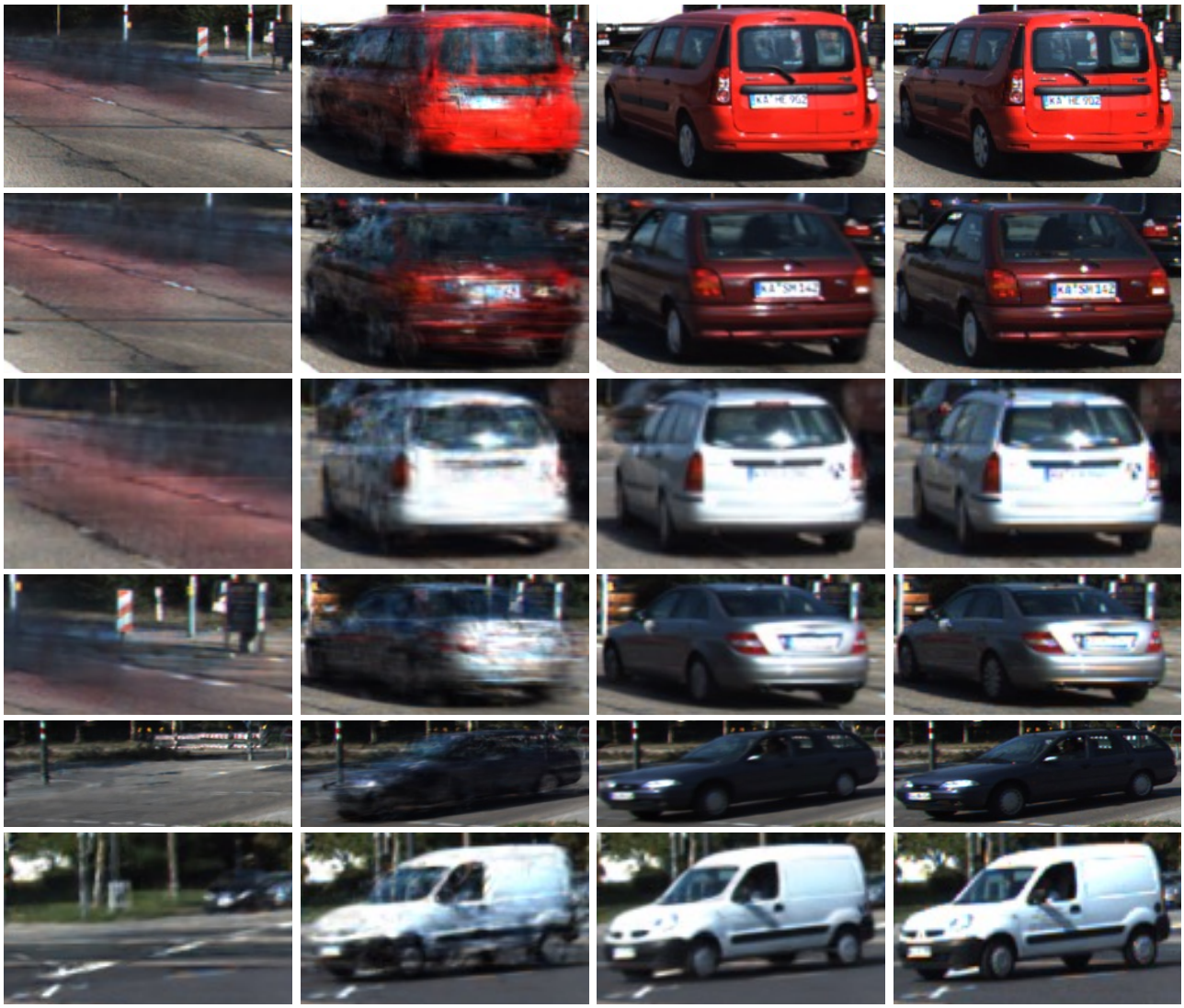
8. Video Demos

We include five video demos in the supplementary material according to size limitation.

Demos 1 and 2 showcase the results of image synthesis from novel views produced by SplatFlow, alongside baseline methods and ground truth (G.T.) data, in dynamic driving scenarios from the Waymo and KITTI datasets respectively. In these videos, the first and second rows display the surrounding images rendered by the baselines: 3D-GS [3], PVG [1], or EmerNeRF [9]. The third row presents the surrounding images rendered by our SplatFlow, while the final row shows the G.T. surrounding images. To provide a clearer comparison in the visualization, video demos 1 and 2 are played at 0.1x speed.

Demos 3 and 4 showcase the rendered images, optical flow, and depth by our SplatFlow in dynamic driving scenarios from the Waymo and KITTI datasets. In these videos, the first row shows the G.T RGB images. The second row displays the rendered RGB images, the third row shows the rendered optical flow, and the fourth row presents the rendered depth, all generated by our SplatFlow. For a clearer visualization, video demos 3 and 4 are played at 0.1x speed.

Demo 5 showcases the 3D motion prediction of LiDAR points within the NMFF. The color wheel in the top right corner represents the flow magnitude through color intensity and flow direction via the angle. For clearer visualization, video Demo 5 is played at 0.1x speed.



3D GS PVG **SplatFlow (ours)** G.T.

Figure 5. Detailed comparison of dynamic object synthesis from novel views on KITTI dataset.

References

[1] Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *arXiv preprint arXiv:2311.18561*, 20. 1, 2, 3

[2] Jianfei Guo, Nianchen Deng, Xinyang Li, Yeqi Bai, Botian Shi, Chiyu Wang, Chenjing Ding, Dongliang Wang, and Yikang Li. Streetsurf: Extending multi-view implicit surface reconstruction to street views. *arXiv preprint arXiv:2306.04988*, 2023. 1, 2

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1, 2, 3

[4] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2021. 1, 2

[5] Jhony Kaesemodel Pontes, James Hays, and Simon Lucey. Scene flow from point clouds with or without learning. In *2020 international conference on 3D vision (3DV)*, pages 261–270. IEEE, 2020. 1

[6] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12375–12385, 2023. 1, 2

[7] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pages 3–15. Springer, 2023. 1

[8] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li

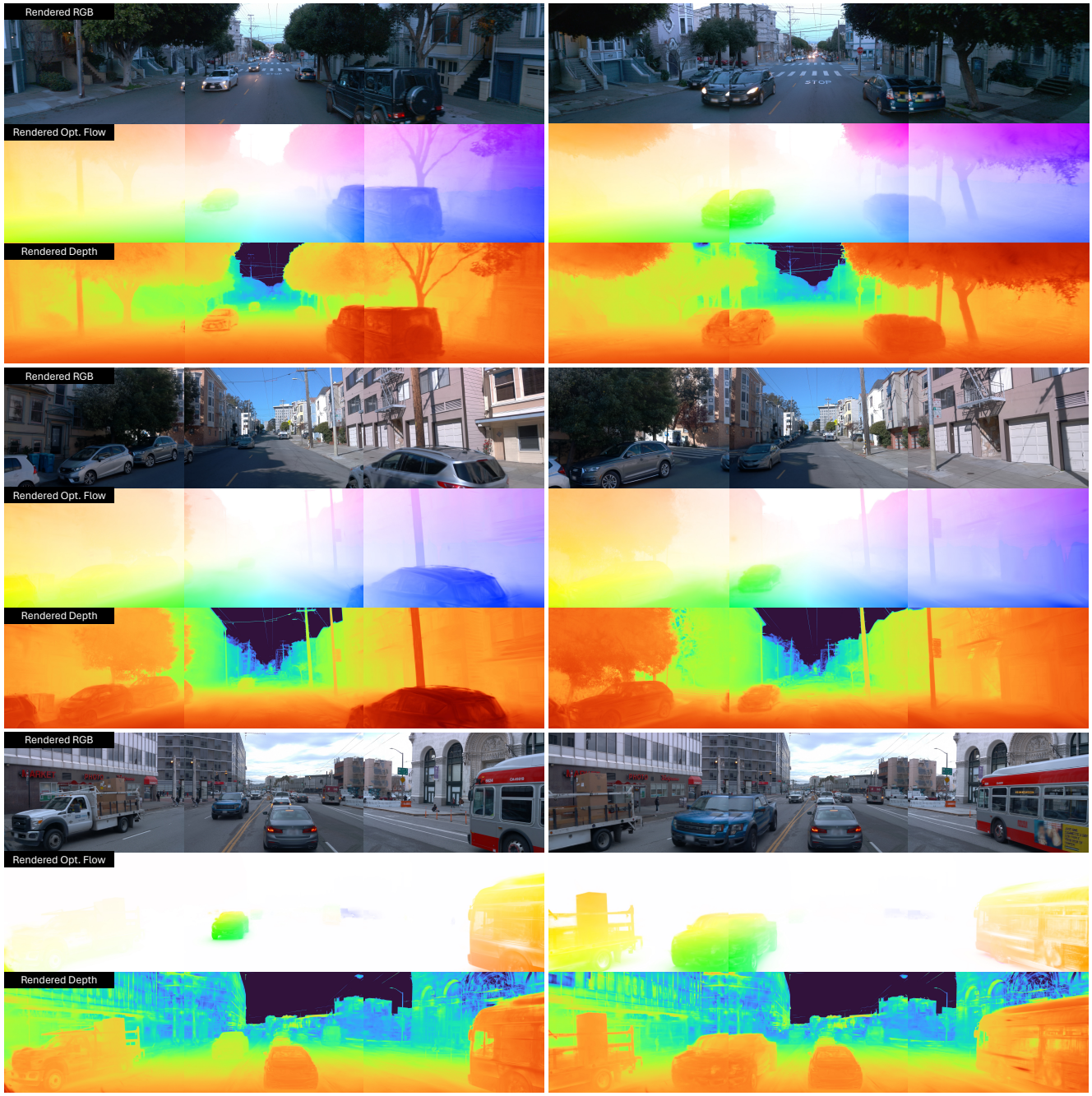


Figure 6. Visualization of rendered RGB image, optical flow, and depth by SlpatFlow on Waymo dataset.

Zhang. S-nerf: Neural radiance fields for street views. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 2

[9] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 1, 2, 3

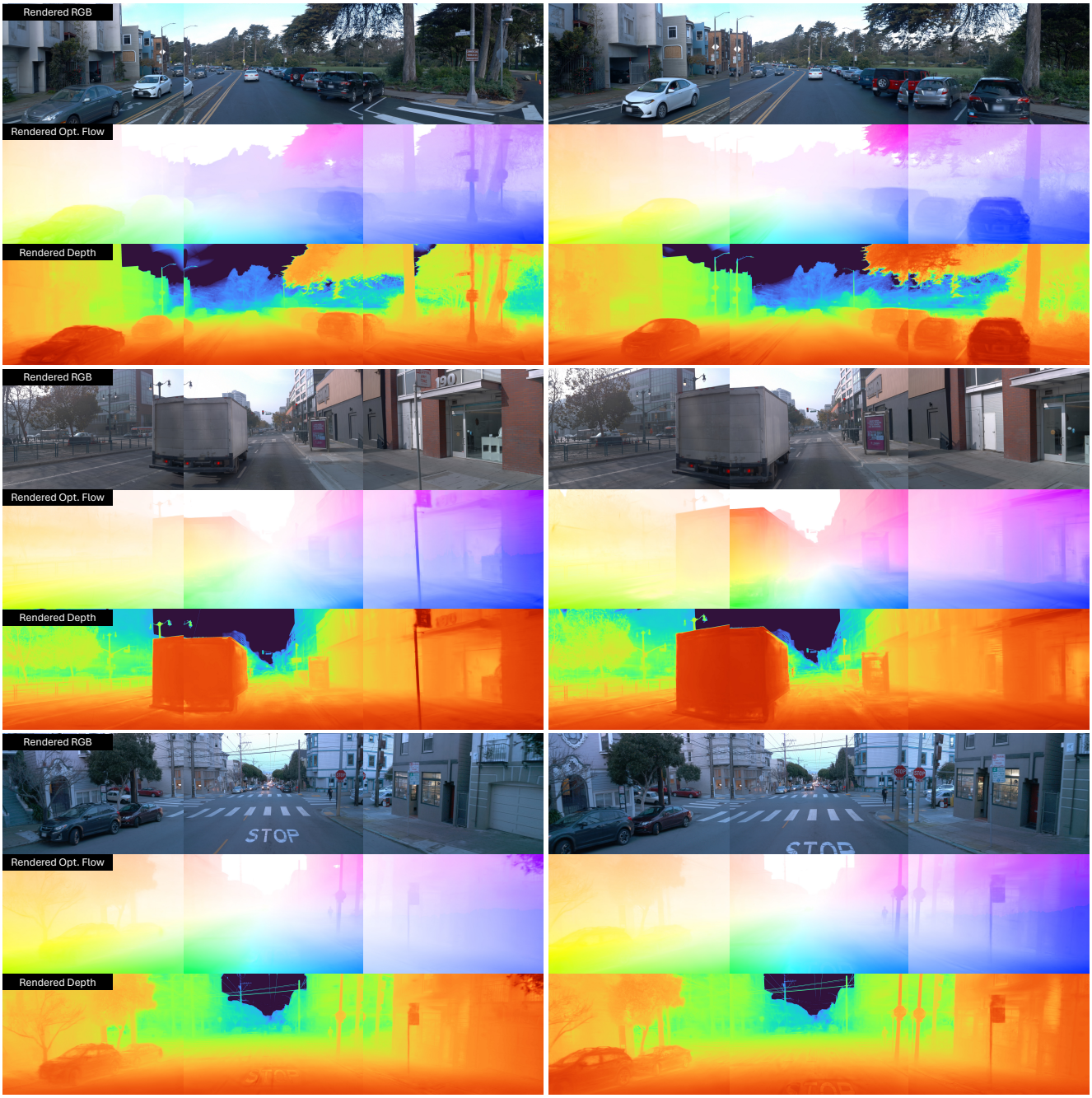


Figure 7. Visualization of rendered RGB image, optical flow, and depth by SplatFlow on Waymo dataset.

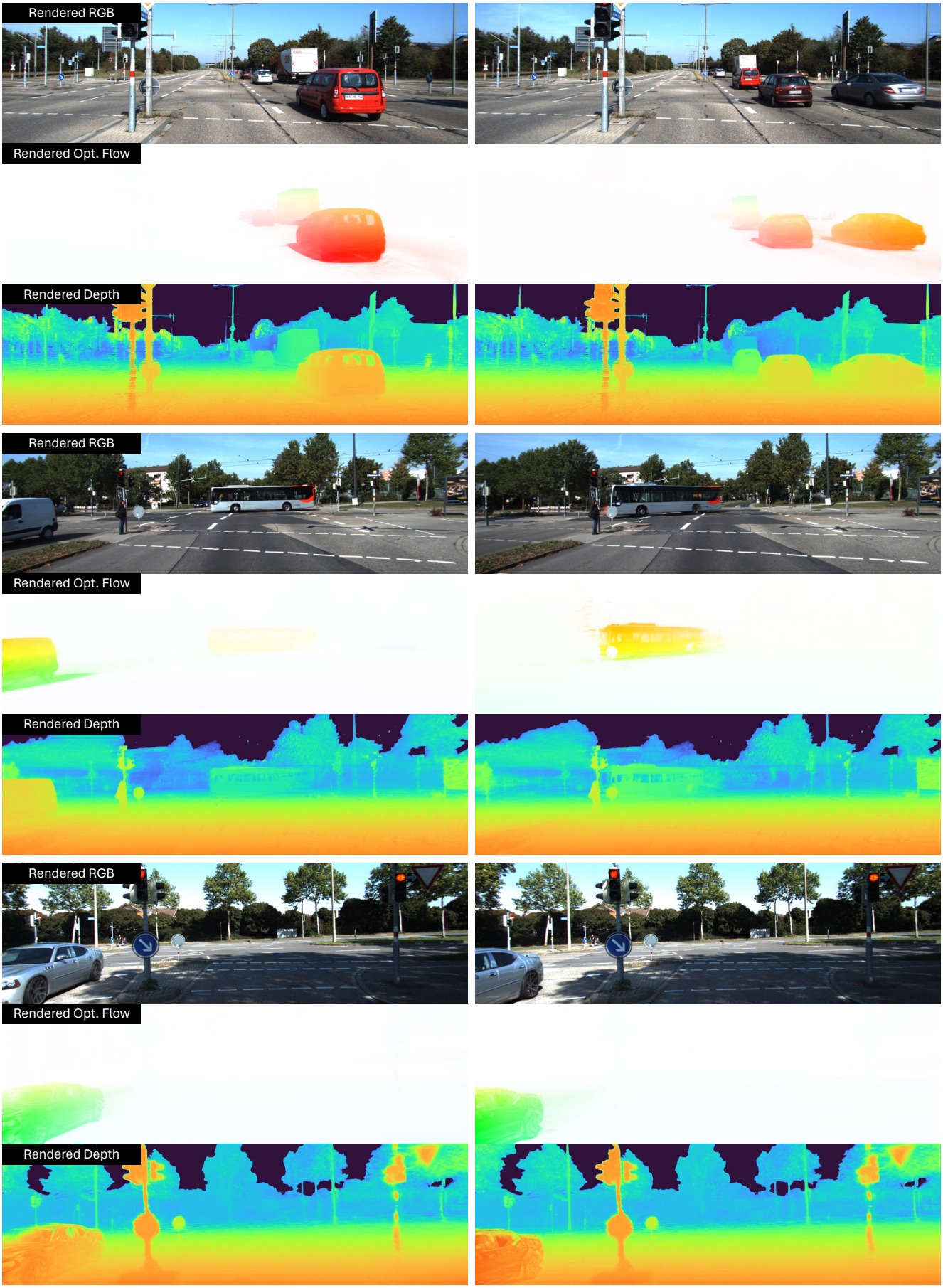


Figure 8. Visualization of rendered RGB image, optical flow, and depth by SlpatFlow on KITTI dataset.