GBG-SLAM: Key-frame Centered Monocular Gaussian SLAM with Scale Consistent Dense Geometry Boosting

Supplementary Material

Abstract

This supplementary material is organized as follows: Section 1 provides the system details and hyperparameters for tracking and mapping. Section 2 presents the complete comparison tables for rendering performance. Section 3 explains the baseline methods. Section 4 discusses the limitations of this work and future directions. Section 5 provides additional qualitative results.

1. System and Hyperparameters

For tracking, we use the pre-trained DROID module [8] to calculate optical flow. The motion filter threshold is set to 2.4 to trigger new keyframe creation. The multi-view filter, used for extracting reliable depth predictions, uses a reprojection threshold of 0.01 for inspecting depth predictions. The local bundle adjustment (BA) window size is set to 25. We perform global optimization every 20 keyframes.

For mapping, the learning rate for 3D Gaussian positions is set to 0.00016 for initialization, and 0.0000016 in the standard loop. The learning rates for spherical harmonics (DC only), opacity, scale, and rotation are fixed at 0.0025, 0.05, 0.001, and 0.001, respectively. The 3D Gaussians are initialized from the first keyframe with 500 iterations of optimization by minimizing the loss defined in Equation **??**. During the standard mapping loop, the number of iterations is set to 30 per step for quick convergence. The optimization occurs in a local keyframe window of size 10 to prevent Gaussian optimization overfitting. Gaussian pruning and densification occur every 150 iterations. Gaussians are pruned if their opacity falls below 0.8, their scale exceeds 0.1, or their 2D screen-space projection size exceeds 20 pixels. Keyframe camera poses are updated during mapping with learning rates of 0.003 for camera rotation and 0.001 for camera translation. The final refinement phase, consisting of 25,000 iterations is, applied at the very end to optimize all 3D Gaussians globally.

2. Rendering performance table

In this section, we show the complete comparison tables for Gaussian rendering performance. Table 1, Table 2 and Table 3 demonstrates the Gaussian rendering performance for Replica [6], ScanNet [1] and TUM-RGBD [7], respectively.

3. Baseline Methods

We compare our methods with state-of-the-art RGB-D Gaussian SLAM methods. Please refer to the complete tables 1, 2 and 3. For visual comparison, we choose the seminal work SplaTAM [3] which sets for the first Gaussian splatting SLAM and the G-SLAM [11] which demonstrates the best performance so far.

4. Limitations

Although our GBG-SLAM shows superior performance in our experiments, the system still has several limitations.

First, the current system uses DROID for tracking, which computes the optical flow via 4D correlations of 2D deep features. Consequently, every keyframe must store these 2D dense feature maps. Due to the memory overhead for storing the dense 2D features, the length of the input video is limited. To overcome this issue, replacing the current pipeline with a sparse (or deep sparse) feature-based tracking module would be a promising direction for achieving a fast and lightweight SLAM system.

Second, although our system achieves faster speeds compared to state-of-the-art methods, it still falls short of real-time operation ($\sim$ 30 frames per second) on the benchmark datasets. Accelerating the overall pipeline to bridge this gap remains an important avenue for future research.

5. Additional Qualitative Results

We provide additional qualitative results on the experimental benchmark datasets. Figures 1 - 7 present these additional visual comparisons.

References

[1] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 4, 5, 6, 7, 8

[2] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17408–17419, 2023. 2, 3

[3] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians

Methods	Metrics	Avg.	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4
NICE-SLAM [12]	PSNR ↑	24.42	22.12	22.47	24.52	29.07	30.34	19.66	22.23	24.94
	SSIM ↑	0.809	0.689	0.757	0.814	0.874	0.886	0.797	0.801	0.856
	LPIPS ↓	0.233	0.330	0.271	0.208	0.229	0.181	0.235	0.209	0.198
Vox-Fusion [10]	PSNR ↑	24.41	22.39	22.36	23.92	27.79	29.83	20.33	23.47	25.21
	SSIM ↑	0.801	0.683	0.751	0.798	0.857	0.876	0.794	0.803	0.847
	LPIPS	0.236	0.303	0.269	0.234	0.241	0.184	0.243	0.213	0.199
Co-SLAM [9]	PSNR ↑	30.24	27.27	28.45	29.06	34.14	34.87	28.43	28.76	30.91
	SSIM ↑	0.939	0.901	0.909	0.932	0.961	0.969	0.938	0.941	0.955
	LPIPS ↓	0.252	0.324	0.294	0.266	0.209	0.196	0.258	0.229	0.236
ESLAM [2]	PSNR ↑	29.08	25.32	27.77	29.08	33.71	30.20	28.09	28.77	29.71
	SSIM ↑	0.929	0.875	0.902	0.932	0.960	0.923	0.943	0.948	0.945
	LPIPS ↓	0.336	0.313	0.298	0.248	0.184	0.228	0.241	0.196	0.204
Point-SLAM [5]	PSNR ↑	35.15	32.40	34.08	35.50	38.26	39.16	33.99	33.48	35.15
	SSIM ↑	0.975	0.874	0.890	0.935	0.910	0.942	0.953	0.948	0.923
	LPIPS ↓	0.124	0.113	0.116	0.111	0.100	0.118	0.156	0.132	0.142
SplaTAM [3]	PSNR ↑	33.98	32.48	33.72	34.96	38.34	39.04	31.90	29.70	31.68
	SSIM ↑	0.969	0.975	0.970	0.982	0.982	0.982	0.965	0.950	0.946
	LPIPS ↓	0.099	0.072	0.096	0.074	0.083	0.093	0.100	0.118	0.155
Gaussian-SLAM [4]	PSNR ↑	34.66	32.50	34.25	35.10	38.54	39.20	32.90	32.05	32.75
	SSIM ↑	0.973	0.976	0.978	0.981	0.984	0.980	0.967	0.966	0.949
	LPIPS	0.096	0.070	0.094	0.070	0.086	0.087	0.101	0.115	0.148
G-SLAM [11]	PSNR ↑	42.08	38.88	41.80	42.44	46.40	45.29	40.10	39.06	42.65
	SSIM ↑	0.996	0.993	0.996	0.998	0.997	0.997	0.997	0.997	0.996
	LPIPS ↓	0.018	0.017	0.018	0.019	0.015	0.016	0.020	0.020	0.020
Ours	PSNR ↑	37.97	36.35	36.36	36.75	42.90	42.00	35.57	35.92	37.89
	SSIM ↑	0.970	0.965	0.965	0.969	0.986	0.983	0.955	0.964	0.975
	LPIPS ↓	0.026	0.029	0.028	0.030	0.011	0.015	0.043	0.023	0.027

Table 1. Rendering performance on the Replica dataset [6]. Our GBG-SLAM outperforms most existing dense neural RGB-D methods across the commonly reported rendering metrics. The best, second-best, and third-best results are highlighted in first, second, third.

for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

[4] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024. 2

[5] Erik Sandström, Yue Li, Luc Van Gool, and Martin R Oswald. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18433–18444, 2023. 2, 3

[6] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 2

[7] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 1, 9, 10

[8] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 1

[9] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13293–13302, 2023. 2

[10] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022. 2, 3

[11] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

[12] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022. 2, 3

Methods	Metrics	Avg.	0000	0059	0106	0169	0181	0207
NICE-SLAM [12]	PSNR ↑	17.54	18.71	16.55	17.29	18.75	15.56	18.38
	SSIM ↑	0.621	0.641	0.605	0.646	0.629	0.562	0.646
	LPIPS ↓	0.548	0.561	0.534	0.510	0.534	0.602	0.552
Vox-Fusion [10]	PSNR ↑	18.17	19.06	16.38	18.46	18.69	16.75	19.66
	SSIM ↑	0.673	0.662	0.615	0.753	0.650	0.666	0.696
	LPIPS ↓	0.504	0.515	0.528	0.439	0.513	0.532	0.500
ESLAM [2]	PSNR ↑	15.29	15.70	14.48	15.44	14.56	14.22	17.32
	SSIM ↑	0.658	0.687	0.632	0.628	0.656	0.696	0.653
	LPIPS ↓	0.488	0.449	0.450	0.529	0.486	0.482	0.534
Point-SLAM [5]	PSNR ↑	19.82	21.30	19.48	16.80	18.53	22.27	20.56
	SSIM ↑	0.751	0.806	0.765	0.676	0.686	0.823	0.750
	LPIPS ↓	0.514	0.485	0.499	0.544	0.542	0.471	0.544
SplaTAM [3]	PSNR ↑	19.14	19.33	19.27	17.73	21.97	16.76	19.8
	SSIM ↑	0.716	0.660	0.792	0.690	0.776	0.683	0.696
	LPIPS ↓	0.358	0.438	0.289	0.376	0.281	0.420	0.341
G-SLAM [11]	PSNR ↑	27.67	28.54	26.21	26.26	28.60	27.79	28.63
	SSIM ↑	0.923	0.926	0.934	0.917	0.922	0.914	0.923
	LPIPS ↓	0.248	0.271	0.211	0.217	0.226	0.277	0.288
Ours	PSNR ↑	31.36	32.86	28.49	30.80	32.70	31.28	31.94
	SSIM ↑	0.903	0.914	0.890	0.921	0.914	0.902	0.879
	LPIPS ↓	0.108	0.068	0.116	0.094	0.085	0.152	0.130

Table 2. Rendering performance on the ScanNet dataset. Our GBG-SLAM achieves the best performance on this real-world dataset. The best, second-best, and third-best results are highlighted in first, second, and third, respectively.

Methods	Metrics	Avg.	fr1/dsk.	fr2/xyz	fr3/of.
NICE-SLAM [12]	PSNR ↑	14.86	13.83	17.87	12.89
	SSIM ↑	0.614	0.569	0.718	0.554
	LPIPS ↓	0.441	0.482	0.344	0.498
Vox-fusion [10]	PSNR ↑	16.46	15.79	16.32	17.27
	SSIM ↑	0.677	0.647	0.706	0.677
	LPIPS ↓	0.471	0.523	0.433	0.456
ESLAM [2]	PSNR ↑	15.26	11.29	17.56	17.02
	SSIM ↑	0.478	0.666	0.310	0.478
	LPIPS ↓	0.569	0.358	0.698	0.652
Point-SLAM [5]	PSNR ↑	16.62	13.87	17.56	18.43
	SSIM ↑	0.696	0.627	0.708	0.754
	LPIPS ↓	0.526	0.544	0.585	0.448
SplaTAM [3]	PSNR ↑	22.80	22.00	24.50	21.90
	SSIM ↑	0.893	0.857	0.947	0.876
	LPIPS ↓	0.202	0.232	0.100	0.202
G-SLAM [11]	PSNR ↑	25.05	24.01	25.02	26.13
	SSIM ↑	0.929	0.924	0.924	0.939
	LPIPS ↓	0.168	0.178	0.186	0.141
Ours	PSNR ↑	28.24	26.16	30.33	28.22
	SSIM ↑	0.883	0.860	0.915	0.875
	LPIPS ↓	0.117	0.156	0.066	0.129

Table 3. Complete rendering performance on the TUM-RGBD dataset. Our GBG-SLAM achieves the best average performance. The best, second-best, and third-best results are highlighted in first, second, and third, respectively.

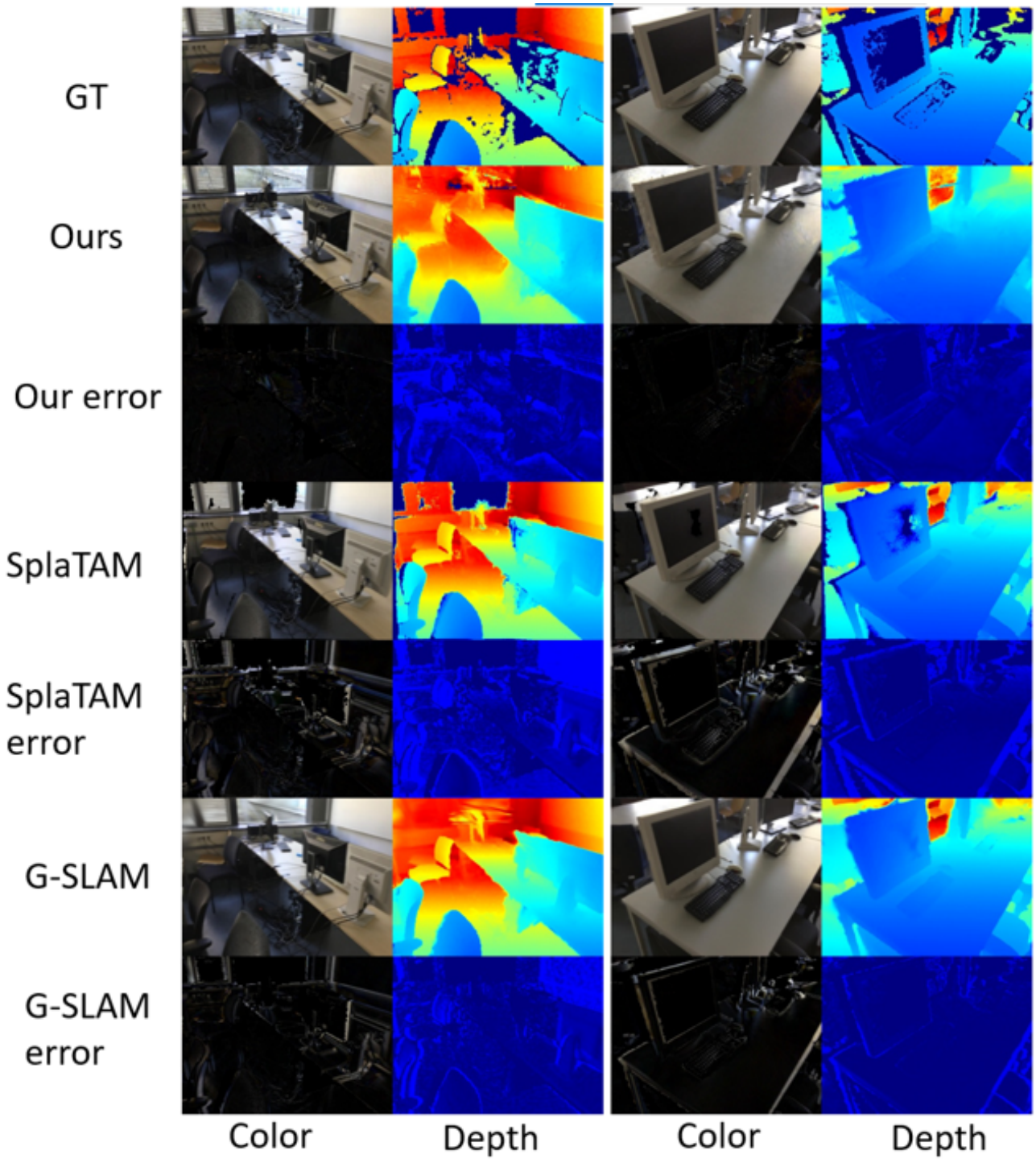

Figure 1. Rendering results on the ScanNet [1] scene0000. The first row shows color images normalized to [0, 1] and corresponding error maps clamped to [0, 0.2]. From left to right: ground truth, our method, our error map, SplaTAM [3], SplaTAM error[3], G-SLAM [11], G-SLAM error.

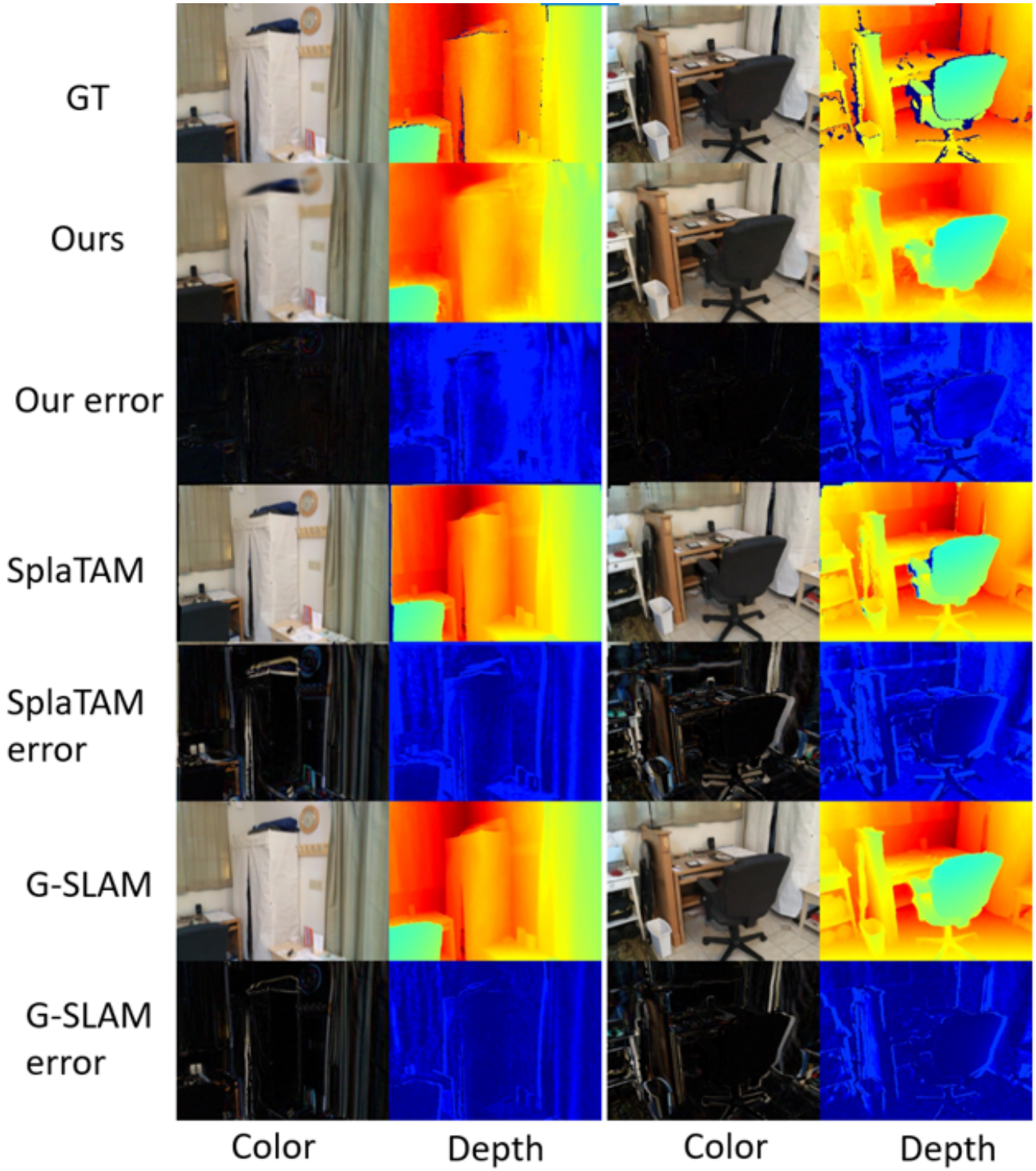

Figure 2. Rendering results on the ScanNet [1] scene0106. The first row shows color images normalized to [0, 1] and corresponding error maps clamped to [0, 0.2]. From left to right: ground truth, our method, our error map, SplaTAM [3], SplaTAM error[3], G-SLAM [11], G-SLAM error.

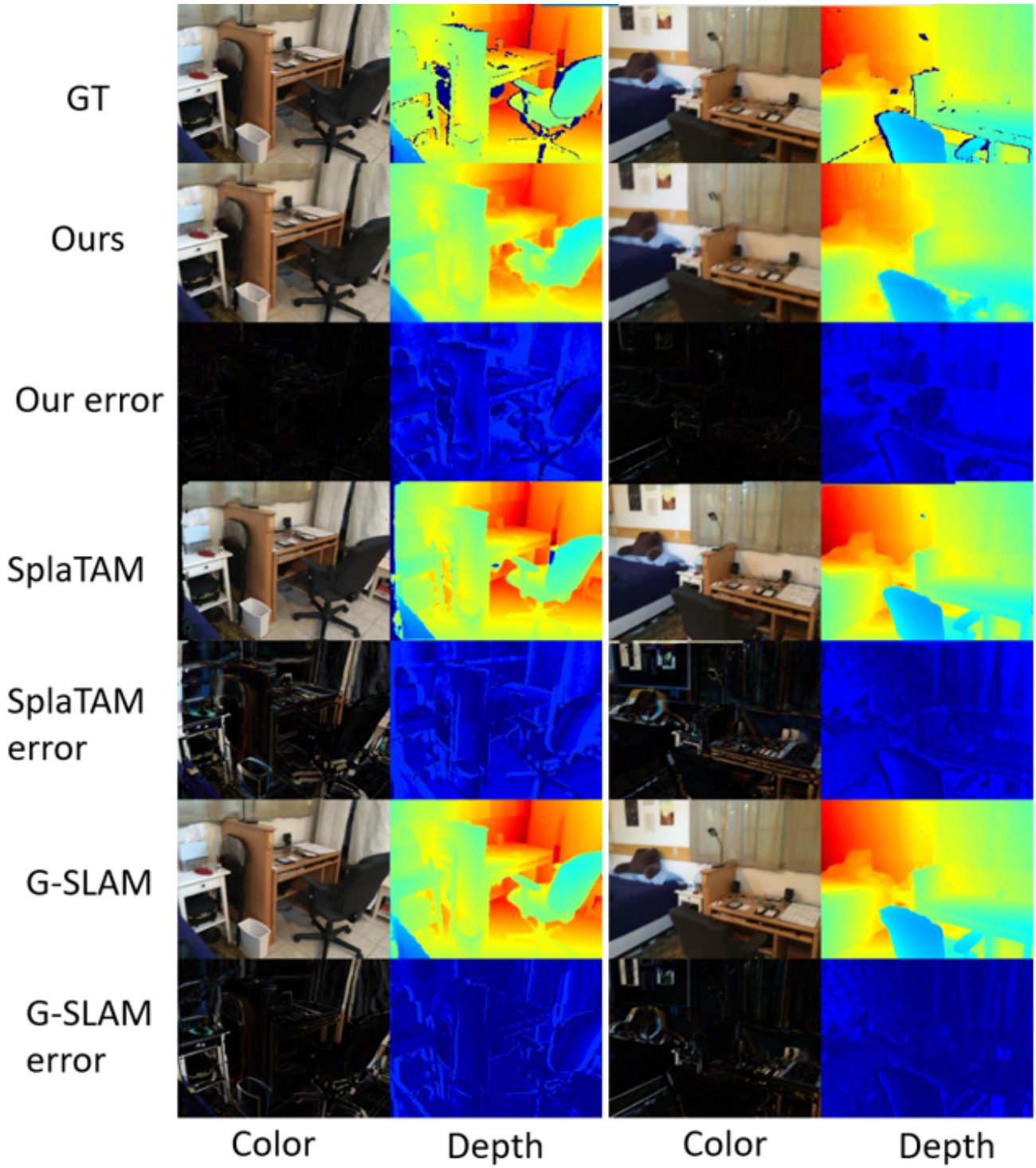

Figure 3. Rendering results on the ScanNet [1] scene0106. The first row shows color images normalized to [0, 1] and corresponding error maps clamped to [0, 0.2]. From left to right: ground truth, our method, our error map, SplaTAM [3], SplaTAM error[3], G-SLAM [11], G-SLAM error.

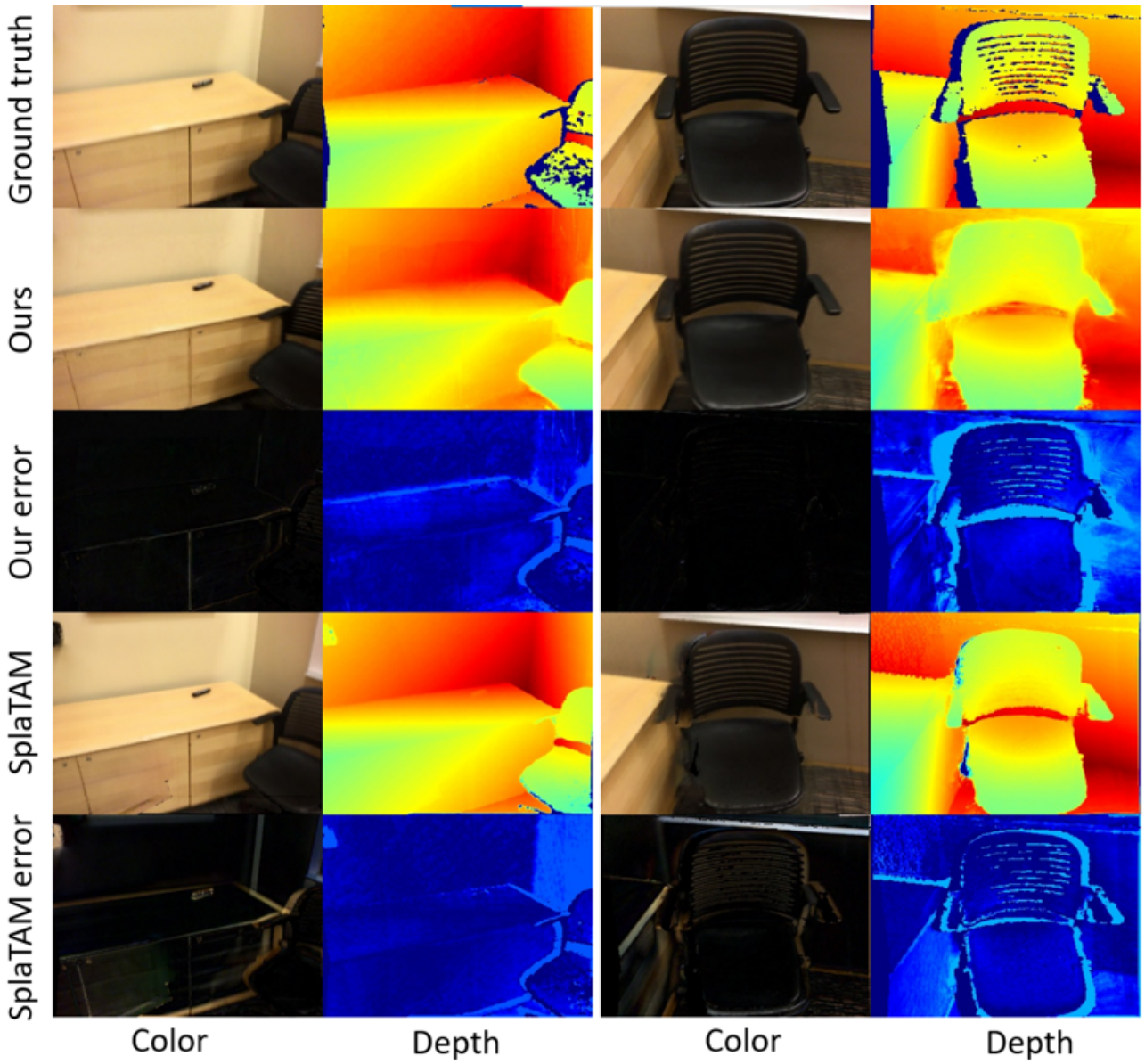

Figure 4. Rendering results on the ScanNet [1] scene0169. The first row shows color images normalized to [0, 1] and corresponding error maps clamped to [0, 0.2]. From left to right: ground truth, our method, our error map, SplaTAM [3], SplaTAM error[3], G-SLAM [11], G-SLAM error.

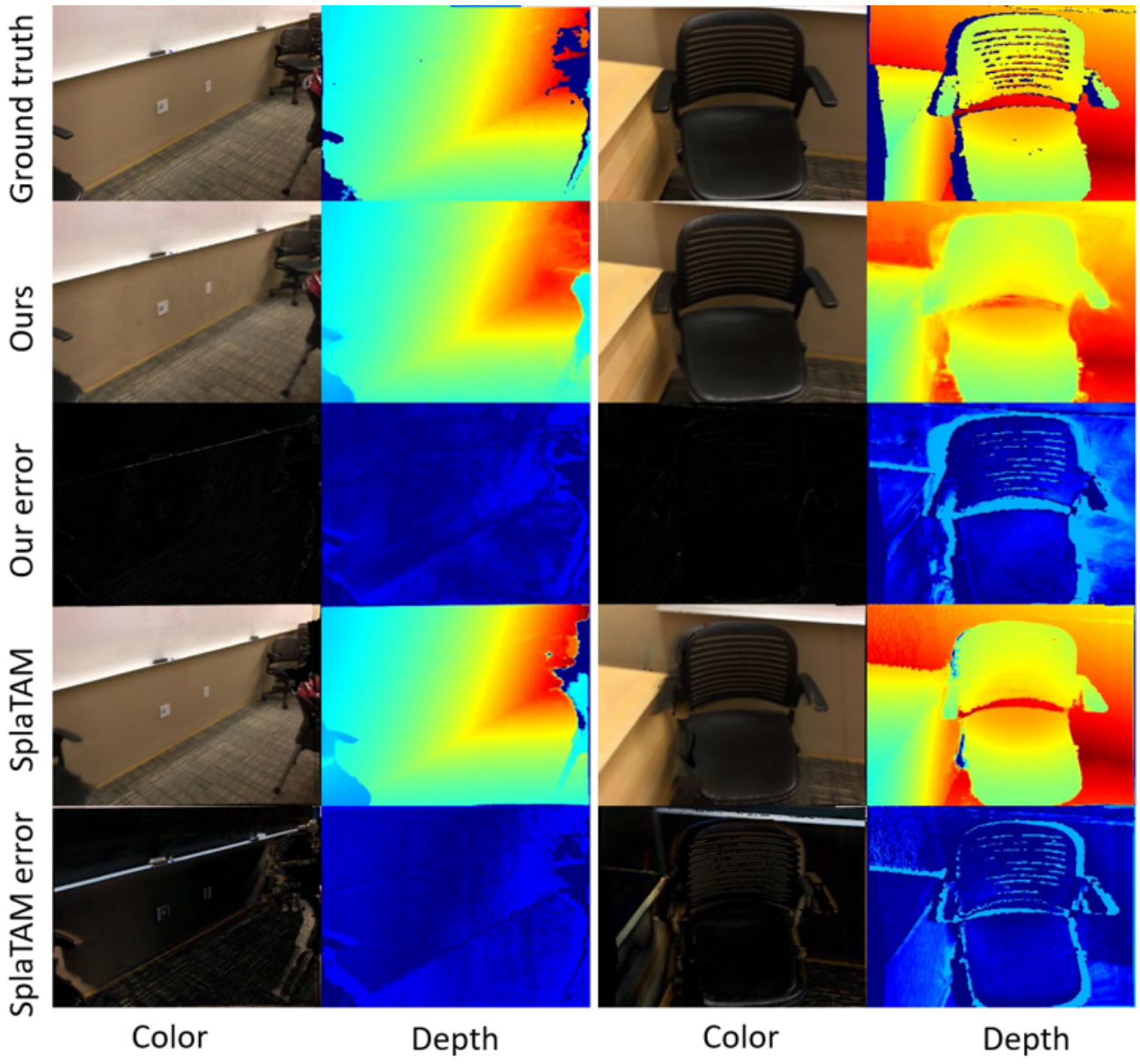

Figure 5. Rendering results on the ScanNet [1] scene0169. The first row shows color images normalized to [0, 1] and corresponding error maps clamped to [0, 0.2]. From left to right: ground truth, our method, our error map, SplaTAM [3], SplaTAM error[3], G-SLAM [11], G-SLAM error.

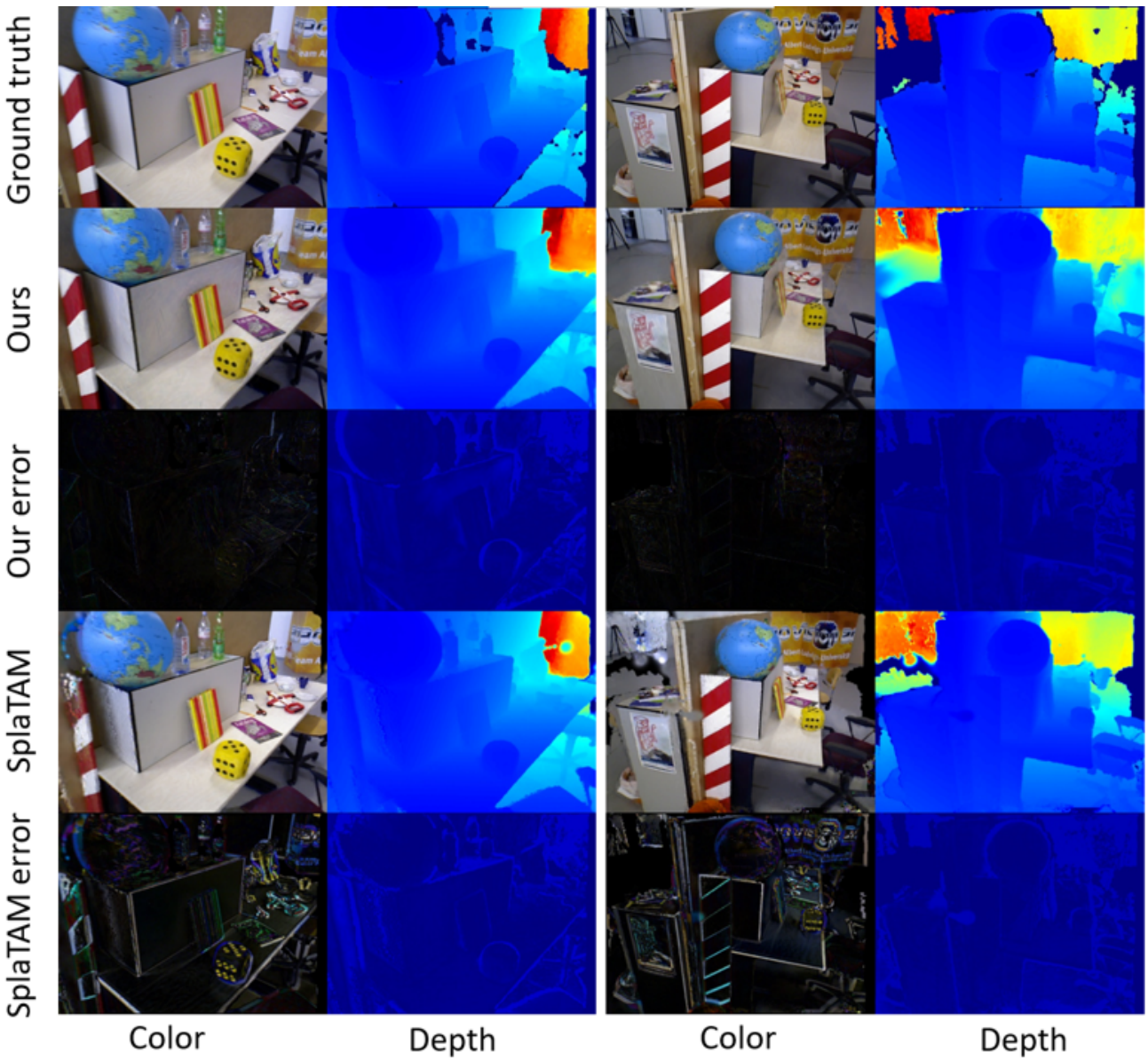

Figure 6. Additional Rendering results of TUM-RGBD dataset freiburg3-office [7]. The first row shows color images normalized to [0, 1] and corresponding error maps clamped to [0, 0.2]. From left to right: ground truth, our method, our error map, SplaTAM [3], SplaTAM error[3], G-SLAM [11], G-SLAM error.

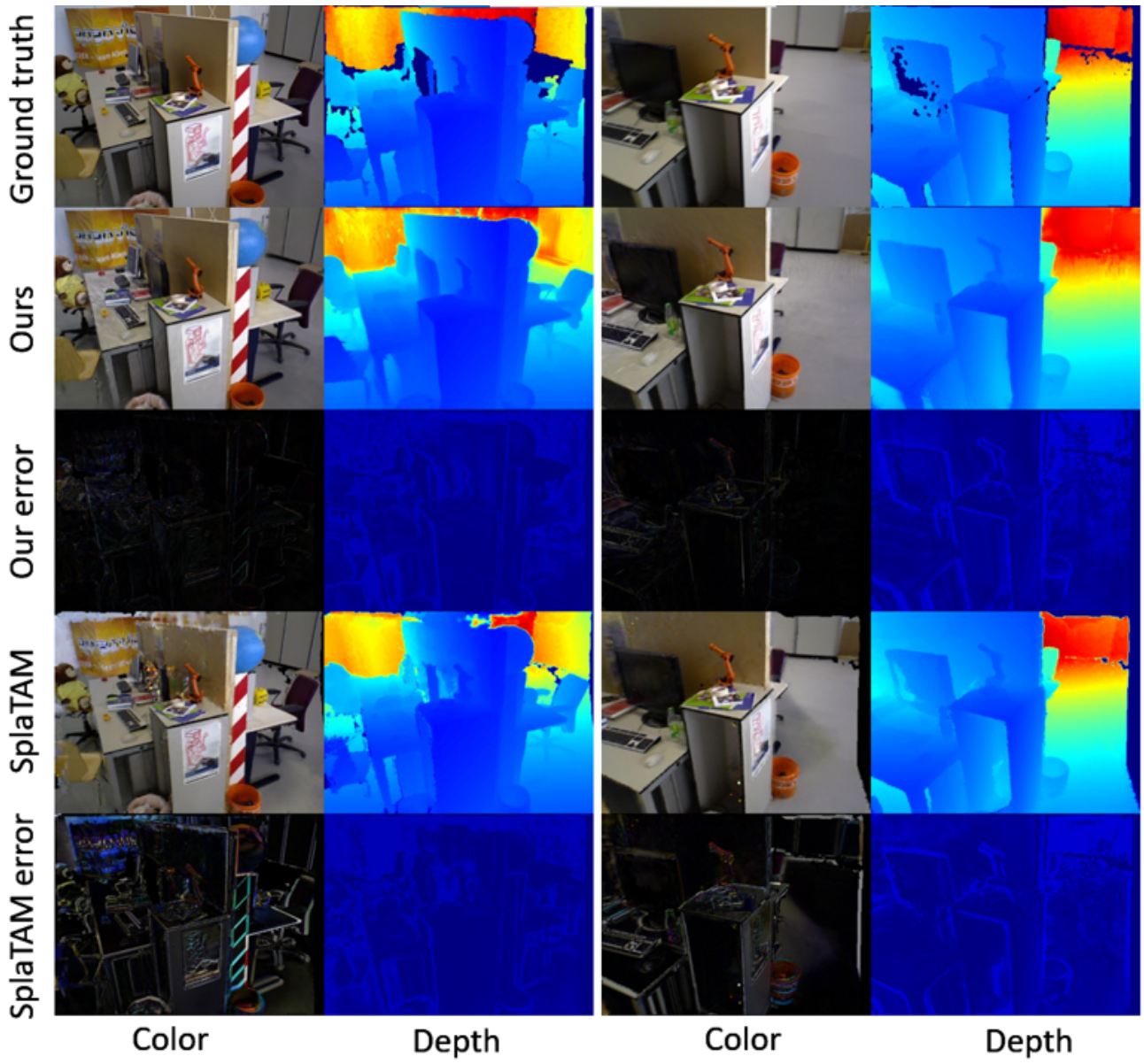

Figure 7. Additional Rendering results of TUM-RGBD dataset freiburg3-office [7]. The first row shows color images normalized to [0, 1] and corresponding error maps clamped to [0, 0.2]. From left to right: ground truth, our method, our error map, SplaTAM [3], SplaTAM error[3], G-SLAM [11], G-SLAM error.