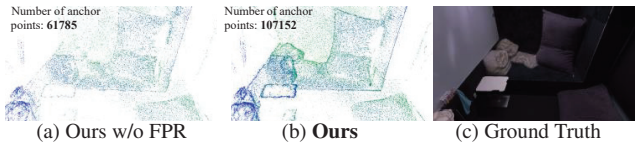
SEGS-SLAM: Structure-enhanced 3D Gaussian Splatting SLAM with Appearance Embedding

Supplementary Material



(a) Ours w/o FPR (b) **Ours** (c) Ground Truth

Figure 10. Visualization of anchor points after 30K iterations. Increasing the number of Gaussians along edges improves the rendering quality. In the figure, 3D Gaussians are $k = 10$ times the number of anchor points.

7. Overview

The supplementary material is organized as follows: (1) Sec. 8 introduces more details of FPR. (2) Sec. 9 presents additional ablation studies. (3) Sec. 10 provides real-time performance for all methods. (4) Sec. 11 provides additional implementation details, including the detailed pipeline for localization and geometry mapping (Sec. 11.2), the MLP architecture used for structured 3D Gaussians (Sec. 11.3), the MLP structure for Appearance-from-Motion Embedding (Sec. 11.4), and anchor point refinement (Sec. 8). (6) Sec. 12 presents quantitative results for each scene and includes more comparative renderings.

8. Details of FPR

In scenes with simple structures, our structured 3D Gaussians can effectively model both structure and appearance changes. However, we observe that structured 3D Gaussians perform poorly in rendering high-frequency details, such as object edges and areas with complex textures. Hence, we propose the frequency pyramid regularization (FPR) technique, which effectively leverages multi-scale frequency spectra. Here, we introduce the frequency pyramid to improve the consistency of rendering details for the same object across varying viewpoint distances. Unlike FreGS [47], we leverage only high-frequency information, as low-frequency components typically represent scene structure, which is already effectively captured by our structured 3D Gaussians as shown in Tab. 6.

The primary effect of FPR is to guide the densification of anchor points. Specifically, when the average gradient of all Gaussians within a voxel exceeds a threshold τ_g, a new anchor point is added at the center of the voxel. Consequently, if a high-frequency region in the scene exhibits a substantial discrepancy between output rendering and ground truth, the total loss includes the frequency regularization $\mathcal{L}_{hf}$ increases, pushing the average gradient beyond τ_g. Then, more new anchors are added in the region, and scene edges become sharper, as shown in Fig. 10. Thus, the high-frequency details in the scene are refined by FPR.

Camera type	RGB-D		Mono		Stereo
Datasets	Replica	TUM R	Replica	TUM R	EuRoC
Scale level	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑
1	39.14	25.82	37.65	23.83	23.58
2	39.29	25.53	37.55	23.77	23.57
4	39.34	26.02	37.51	24.75	23.42
3 (**Ours**)	**39.42**	**26.03**	**37.96**	**25.17**	**23.64**

Table 5. Ablation Study on the scale level of FPR.

Camera type	RGB-D		Mono		Stereo
Datasets	Replica	TUM R	Replica	TUM R	EuRoC
Metric	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑
low freq.	38.85	25.76	36.79	24.79	23.37
low & high freq.	39.23	25.84	37.72	24.84	23.27
high freq. (**Ours**)	**39.42**	**26.03**	**37.96**	**25.17**	**23.64**

Table 6. Ablation study on the low frequency component of FPR.

9. Additional Ablation Studies

Scale level of FPR. We think that multiple scales improve the consistency under varying observation distances. The scale level of FPR is set to 3. Results are shown in Tab. 5.

The low-frequency component in FPR. In our experiments, we find that the low-frequency component of FPR conflicts with structured 3D Gaussians, resulting in degradation. The best result is using only the high-frequent component in FPR, as shown in Tab. 6.

Replacement of key components. We train two additional models: one replaces AfME with the appearance embeddings (AE) from *Scaffold*-GS [23], and another replaces FPR with the single-scale frequency regularization (SFR). Our AfME differs from AE in two primary ways: 1) AfME uses camera poses as input, whereas AE uses camera indices; 2) AfME employs an MLP network structure, while AE utilizes an embedding layer. SFR refers to using only the original-scale image frequencies in Eq. (9). The distinction between FPR and SFR lies in the use of multi-scale image frequencies in FPR. As shown in Tab. 4, rows (1), (2), and (3), our full method (3) achieves the highest PSNR scores. This demonstrates that, compared with AE, our AfME is more effective in predicting appearance variations across a wide range of novel views, thus avoiding additional training on the test set. On the other hand, it also highlights that by introducing the frequency pyramid, the model maintain consistency in scene details across varying viewpoint distances, leading to superior rendering quality.

Camera type		RGB-D		Mono		Stereo
Datasets		Replica	TUM R	Replica	TUM R	EuRoC
#	Method	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑
(1)	replace AfME with AE	38.22	19.56	36.09	19.33	18.39
(2)	replace FPR with SFR	39.14	25.82	37.65	23.83	23.58
(3)	**Ours**	**39.42**	**26.03**	**37.96**	**25.17**	**23.64**

Table 7. Ablation Study on the **replacement of key components**.

Metric (RGB-D)	MonoGS	Photo-SLAM (-30K)	RTG-SLAM	SplaTAM	SGS-SLAM	GS-ICP SLAM	**Ours**
Rendering FPS ↑	706	**1562** (1439)	447	531	486	630	400
Tracking FPS ↑	1.33	30.30 (**30.87**)	17.24	0.15	0.14	30.32	17.18
Mapping Time ↓	37m40s	**1m20s** (6m32s)	12m03s	3h45m	4h05m	1m32s	11m14s

Table 8. Real-time performance.

10. Real-time performance

Our method, following Photo-SLAM, employs two parallel threads: *Localization & Geometry Mapping* and *3D-GS Mapping*. We note that only tracking and rendering are real-time. The runtime of all methods is provided in Tab. 8.

11. Implementation details

11.1. System Overview

Our system comprises two main modules: *localization and geometry mapping* and *progressively refined 3D Gaussian splatting (3D-GS)*. In our implementation, these two modules run in separate threads. The localization and geometry mapping module focuses on camera pose estimation and scene point cloud mapping. The progressively refined 3D-GS module takes the estimated keyframe poses and point clouds from the localization and geometry mapping module. Then the module incrementally completes the photorealistic mapping of the scene.

11.2. Localization and Geometry Mapping

In our implementation, the localization and geometric mapping module consists of three main threads: *tracking*, *local mapping*, and *loop closing*, along with an on-demand thread for *global bundle adjustment (BA)*. Specifically, the tracking thread performs a motion-only BA to optimize camera poses. The local mapping thread optimizes keyframe poses and map point clouds within a local sliding window via local BA. Lastly, the loop closing thread continuously checks for loop closures. If a loop is detected, a global BA is triggered to jointly optimize the camera poses of all keyframes and all points of the scene.

Motion-only BA. We optimize the camera orientation $\mathbf{R} \in \mathrm{SO}(3)$ and position $t \in \mathbb{R}^3$ through motion-only BA. The camera poses $(\mathbf{R}_\iota, \mathbf{t}_\iota)$ are optimized by minimizing the reprojection error between the matched 3D points $\mathbf{P}_\iota \in \mathbb{R}^3$ and 2D feature points $\mathbf{p}_\iota$ within a sliding window:

$$\{\mathbf{R}_\iota, \mathbf{t}_\iota\} = \sum_{\iota \in \mathcal{X}} \underset{\mathbf{R}_\iota, \mathbf{t}_\iota}{\operatorname{argmin}} \rho(\|p_\iota - \pi(\mathbf{R}_\iota \mathbf{P}_\iota + \mathbf{t}_\iota)\|^2_{\Sigma_g}) \quad (12)$$

where $\mathcal{X}$ represent the set of all matches, Σ_g denote the covariance matrix associated with the keypoint's scale, π is the projection function, and ρ is the robust Huber cost function.

Local BA. We perform a local BA by optimizing a set of covisible keyframes $\mathcal{K}_L$ alone with the set of points P_L observed in those keyframes as follows:

$$\{\mathbf{P}_m, \mathbf{R}_l, \mathbf{t}_l\} = \underset{\mathbf{P}_m, \mathbf{R}_l, \mathbf{t}_l}{\operatorname{argmin}} \sum_{\kappa \in \mathcal{K}_L \cup \mathcal{K}_F} \sum_{j \in \mathcal{X}_k} \rho(E(\kappa, j)) \quad (13)$$

$$E(\kappa, j) = \|\mathbf{p}_j - \pi(\mathbf{R}_\kappa \mathbf{P}_j + \mathbf{t}_\kappa)\|^2_{\Sigma_g} \quad (14)$$

where $m \in P_L$, $l \in \mathcal{K}_L, \mathcal{K}_F$ are all other keyframes, $\mathcal{X}_k$ is the set of matches between keypoints in a keyframe κ and points in P_L.

Global BA. Global BA is a special case of local BA, where all keyframes and map points are included in the optimization, except the origin keyframe, which is kept fixed to prevent gauge freedom.

11.3. Structured 3D Gaussians

MLPs as feature decoders. Following [23], we employ four MLPs as decoders to derive the parameters of each 3D Gaussian, including the opacity MLP M_α, the color MLP M_C, and the covariance MLP M_q, M_s. Each MLP adopts a linear layer followed by ReLU and another linear layer. The outputs are activated by their respective activation functions to obtain the final parameters of each 3D Gaussian. The detailed architecture of these MLPs is illustrated in Fig. 11. In our implementation, the hidden layer dimensions of all MLPs are set to 32.

- For *opacity*, we use $\mathrm{Tanh}(\cdot)$ to activate the output of the final linear layer. Since the opacity values of 3D Gaussians are typically positive, we constrain the value range to $[0, 1)$ to ensure valid outputs.
- For *color*, we use $\mathrm{Sigmoid}$ function to activate the output of the final linear layer, which constrains the color value into a range of $[0, 1)$.
- For *rotation*, following 3D-GS [17], we employ a normalization to activate the output of the final linear layer, ensuring the validity of the quaternion representation for rotation.
- For *scaling*, a $\mathrm{Sigmoid}$ function is applied to activate the output of the final linear layer. Finally, the scaling of each 3D Gaussian is determined by adjusting the scaling l_v of its associated anchor based on the MLP's output, as formulated below:

$$\{s_0, \ldots, s_{k-1}\} = M_s(\hat{f}_v,\ \delta_{vc},\ \vec{\mathbf{d}}_{vc}) \cdot l_v \quad (15)$$

11.4. Appearance-from-Motion Embedding

MLP as feature encoder. For AfME, we employ an MLP M_{θ_a} as the encoder. The input to this MLP is the camera pose corresponding to each image. The MLP M_{θ_a} extracts

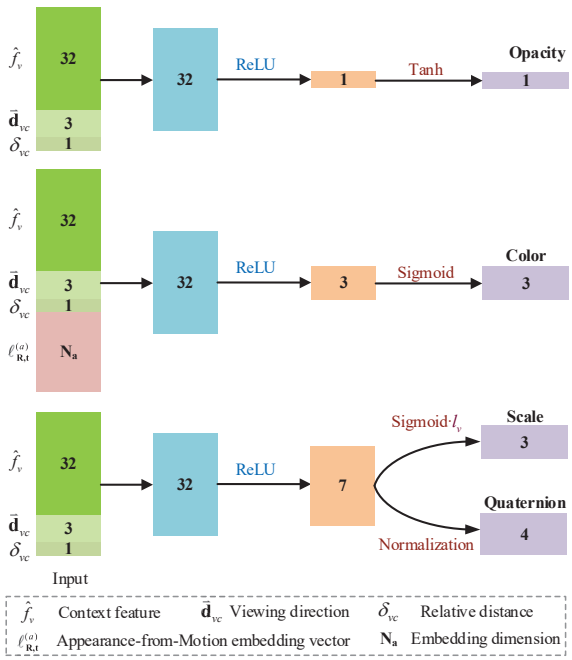


Figure 11. **Structure of the MLPs** M_α, M_C, M_s, and M_q. For each anchor, we use these MLPs to predict the opacity, color, scale, and quaternion of k 3D Gaussian. The inputs to the MLPs include the relative distance δ_{vc} and the viewing direction $\vec{\mathbf{d}}_{vc}$ between camera position $\mathbf{t}_c$ and an anchor point. Since N_a is not a fixed parameter, its specific value is not included in the figure.

pose features and feeds these features to the color decoder M_C. The MLP M_{θ_a} adopts a structure of a linear layer followed by a linear activation function, as illustrated in Fig. 12. The entire pipeline for obtaining the Gaussian color is also detailed in Fig. 12. We adopt an encoder-decoder architecture, where an encoder MLP M_{θ_a} extracts features from the camera poses. Unlike FreGS [47], we leverage only high-frequency information, as low-frequency components typically represent scene structure, which is already effectively captured by our structured 3D Gaussians.

11.5. Anchor Points Refinement

Our anchor refinement strategy follows [23], and it is included here to enhance the completeness of this paper.

Anchor Growing. 3D Gaussians are spatially quantized into voxels of size $\epsilon_g = 0.001$. For all 3D Gaussians within each voxel, we compute the average gradient after $N_t = 100$ training iterations, denoted as ∇_g. When the average gradient ∇_g within a voxel exceeds a threshold $\tau_g = 0.0002$, a new anchor is added at the center of the voxel. Since the total loss includes the frequency regularization $\mathcal{L}_{hf}$ in Eq. (9), anchor points grow toward underrepresented high-frequency regions in the scene. Ultimately, the local details of the scene are refined. In our implementation, the scene is quantized into a multi-resolution voxel grid, allowing new anchors to be added to regions of varying sizes, as defined by

$$\epsilon_g^{(m)} = \epsilon_g / 4^{m-1}, \ \tau_g^{(m)} = \tau_g \cdot 2^{m-1} \tag{16}$$

where m denotes the level of quantization. Additionally, we adopt a random candidate pruning strategy to moderate the growth rate of anchors.

To eliminate redundant anchor points, we evaluate their opacity. Specifically, after N_t training iterations, we accumulate the opacity values of each 3D Gaussian. If the accumulated value α_p falls below a pre-defined threshold, the Gaussian is removed from the scene.

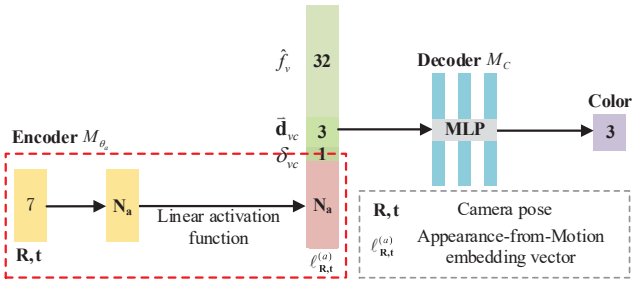


Figure 12. **Structure of the MLPs** M_{θ_a} and M_C. We adopt an encoder-decoder architecture, where an encoder MLP M_{θ_a} first extracts features from the camera poses. For each anchor, the feature $\ell^{(a)}_{\mathbf{R},\mathbf{t}}$, context feature $\hat{f}_v$, the relative distance δ_{vc} between camera position $\mathbf{t}_c$ and the anchor point, and their viewing direction $\vec{\mathbf{d}}_{vc}$ are then fed into a decoder M_C to predict the color of each Gaussian.

11.6. Experimental Parameters

For the monocular camera in the Replica dataset, the dimension of AfME N_a is set to 1, while for other configurations, it is set to 32. Each anchor manages $k = 10$ 3D Gaussians. Anchors with an opacity value below 0.005 are removed. The loss weights λ, λ_{vol}, and λ_{hf} are set to 0.2, 0.01, and 0.01, respectively. For the monocular camera in the Replica dataset, the weight λ_{hf} for frequency regularization is set to 0.025, and the frequency pyramid consists of 3 levels.

12. Additional Qualitative Results

12.1. Per-scene Results.

Tab. 9, Tab. 10, Tab. 13a, Tab. 11, Tab. 12, and Tab. 13b present the photorealistic mapping and localization results of our method across all datasets for each scene. Additionally, Fig. 13, Fig. 14, Fig. 15, Fig. 16, and Fig. 17 show more rendering comparisons between our method and all baseline methods for each scene.

Datasets		Replica									TUM RGB-D			
Method	Metric	R0	R1	R2	Of0	Of1	Of2	Of3	Of4	Avg.	fr1/d	fr2/x	fr3/o	Avg.
MonoGS [26]	PSNR↑	34.29	35.77	36.79	40.87	40.73	35.22	35.89	34.98	36.81	23.59	24.46	24.29	24.11
	SSIM↑	0.953	0.957	0.965	0.979	0.977	0.961	0.962	0.955	0.964	0.783	0.789	0.829	0.800
	LPIPS↓	0.071	0.078	0.074	0.048	0.052	0.074	0.061	0.092	0.069	0.244	0.227	0.223	0.231
Photo-SLAM [14]	PSNR↑	32.09	34.15	35.91	38.70	39.53	33.13	34.15	36.35	35.50	20.14	22.15	20.68	20.99
	SSIM↑	0.920	0.941	0.959	0.967	0.964	0.943	0.943	0.956	0.949	0.722	0.765	0.721	0.736
	LPIPS↓	0.069	0.055	0.041	0.048	0.045	0.075	0.064	0.053	0.056	0.258	0.169	0.211	0.213
Photo-SLAM-30K	PSNR↑	31.41	35.84	38.41	40.44	41.06	34.56	35.43	38.36	36.94	21.78	21.57	21.84	21.73
	SSIM↑	0.873	0.955	0.971	0.975	0.972	0.952	0.954	0.967	0.952	0.766	0.755	0.751	0.757
	LPIPS↓	0.046	0.036	0.026	0.033	0.033	0.059	0.049	0.036	0.040	0.212	0.182	0.165	0.186
RTG-SLAM [29]	PSNR↑	28.49	31.27	32.96	37.32	36.12	31.14	31.19	33.81	32.79	13.62	17.08	18.70	16.47
	SSIM↑	0.834	0.902	0.927	0.957	0.943	0.923	0.918	0.937	0.918	0.501	0.573	0.648	0.574
	LPIPS↓	0.152	0.119	0.122	0.084	0.103	0.145	0.139	0.125	0.124	0.557	0.403	0.422	0.461
GS-SLAM* [42]	PSNR↑	31.56	32.86	32.59	38.70	41.17	32.36	32.03	32.92	34.27	-	-	-	-
	SSIM↑	0.968	0.973	0.971	**0.986**	**0.993**	**0.978**	**0.970**	0.968	**0.975**	-	-	-	-
	LPIPS↓	0.094	0.075	0.093	0.050	0.033	0.094	0.110	0.112	0.082	-	-	-	-
SplaTAM [16]	PSNR↑	32.54	33.58	35.03	38.00	38.85	31.71	29.74	31.40	33.85	21.02	23.39	19.81	21.41
	SSIM↑	0.938	0.936	0.952	0.963	0.955	0.928	0.902	0.914	0.936	0.753	0.806	0.731	0.764
	LPIPS↓	0.068	0.096	0.072	0.087	0.095	0.100	0.119	0.157	0.099	0.341	0.204	0.249	0.265
SGS-SLAM [19]	PSNR↑	32.48	33.50	35.11	38.22	38.91	31.86	30.05	31.53	33.96	-	-	-	-
	SSIM↑	**0.975**	0.968	**0.983**	0.983	0.982	0.966	0.952	0.946	0.969	-	-	-	-
	LPIPS↓	0.071	0.099	0.073	0.083	0.091	0.099	0.118	0.154	0.099	-	-	-	-
GS-ICP SLAM [11]	PSNR↑	34.89	37.15	37.89	41.62	42.86	32.69	31.45	38.54	37.14	15.67	18.49	19.25	17.81
	SSIM↑	0.955	0.965	0.970	0.981	0.981	0.965	0.959	0.969	0.968	0.574	0.667	0.692	0.642
	LPIPS↓	0.048	0.045	0.047	0.027	0.031	0.057	0.057	0.045	0.045	0.444	0.308	0.329	0.361
Ours	PSNR↑	**37.07**	**39.54**	**40.33**	**42.04**	**43.21**	**36.38**	**37.18**	**39.62**	**39.42**	**25.29**	**26.35**	**26.46**	**26.03**
	SSIM↑	0.968	**0.977**	0.980	0.982	0.979	0.967	0.969	**0.977**	**0.975**	**0.839**	**0.831**	**0.859**	**0.843**
	LPIPS↓	**0.023**	**0.016**	**0.015**	**0.020**	**0.019**	**0.035**	**0.026**	**0.018**	**0.021**	**0.136**	**0.081**	**0.105**	**0.107**

Table 9. Quantitative evaluation of our method compared to state-of-the-art methods for **RGB-D** camera on Replica and TUM RGB-D datasets. The best results are marked as **best score**, second best score and third best score. GS-SLAM* denotes the result of GS-SLAM taken from [42], and all others are obtained in our experiments. '-' denotes that the system does not provide valid results.

Datasets (Camera)		Replica (**Mono**)									TUM RGB-D (**Mono**)			
Method	Metric	R0	R1	R2	Of0	Of1	Of2	Of3	Of4	Avg.	fr1/d	fr2/x	fr3/o	Avg.
MonoGS [26]	PSNR↑	26.19	25.42	27.83	31.90	34.22	26.09	28.56	26.49	28.34	20.38	21.21	21.41	21.00
	SSIM↑	0.819	0.798	0.889	0.911	0.930	0.881	0.898	0.897	0.878	0.691	0.690	0.735	0.705
	LPIPS↓	0.246	0.368	0.252	0.249	0.192	0.268	0.189	0.284	0.256	0.377	0.377	0.426	0.393
Photo-SLAM [14]	PSNR↑	30.43	32.11	32.89	37.24	38.10	31.60	32.27	34.16	33.60	19.56	20.82	20.12	20.17
	SSIM↑	0.890	0.926	0.937	0.960	0.955	0.932	0.928	0.943	0.934	0.705	0.718	0.702	0.708
	LPIPS↓	0.099	0.073	0.069	0.062	0.061	0.094	0.084	0.073	0.077	0.281	0.158	0.233	0.224
Photo-SLAM-30K	PSNR↑	32.13	33.14	37.27	38.04	41.73	35.22	34.88	36.22	36.08	22.57	20.54	20.08	21.06
	SSIM↑	0.896	0.921	0.965	0.964	**0.974**	0.952	0.949	0.955	0.947	0.787	0.714	0.697	0.733
	LPIPS↓	0.056	0.086	0.035	0.055	**0.033**	0.061	0.057	0.052	0.054	0.179	0.166	0.213	0.186
Ours	PSNR↑	**34.94**	**37.96**	**38.28**	**41.19**	**42.23**	**36.30**	**35.44**	**37.33**	**37.96**	**23.94**	**25.39**	**26.17**	**25.17**
	SSIM↑	**0.949**	**0.967**	**0.971**	**0.978**	0.972	**0.964**	**0.952**	**0.959**	**0.964**	**0.804**	**0.813**	**0.857**	**0.825**
	LPIPS↓	**0.039**	**0.027**	**0.026**	**0.027**	0.036	**0.038**	**0.055**	**0.050**	**0.037**	**0.135**	**0.121**	**0.110**	**0.122**

Table 10. Quantitative evaluation of our method compared to state-of-the-art methods for **Monocular (Mono)** camera on Replica and TUM RGB-D datasets. The best results are marked as **best score** and second best score.

Datasets		Replica									TUM RGB-D			
Method	Metric	R0	R1	R2	Of0	Of1	Of2	Of3	Of4	Avg.	fr1/d	fr2/x	fr3/o	Avg.
ORB-SLAM3 [3]	RMSE↓	0.500	0.537	0.731	0.762	1.338	0.636	0.419	9.319	1.780	5.056	0.390	1.143	2.196
DRIOD-SLAM [37]	RMSE↓	95.994	52.471	62.908	54.807	36.038	118.191	94.200	79.510	74.264	36.057	16.749	169.844	74.216
MonoGS [26]	RMSE↓	0.444	0.273	0.274	0.442	0.469	0.220	0.159	2.237	0.565	**1.531**	1.440	1.535	1.502
Photo-SLAM [14]	RMSE↓	0.529	0.397	0.295	0.501	0.379	1.202	0.768	0.585	0.582	3.578	**0.337**	1.696	1.870
RTG-SLAM [29]	RMSE↓	0.222	0.258	0.248	**0.201**	**0.190**	**0.115**	**0.156**	0.136	0.191	1.582	0.377	**0.996**	**0.985**
GS-SLAM* [42]	RMSE↓	0.480	0.530	0.330	0.520	0.410	0.590	0.460	0.700	0.500	3.300	1.300	6.600	3.700
SplaTAM [16]	RMSE↓	0.501	0.220	0.298	0.316	0.582	0.256	0.288	0.279	0.343	5.102	1.339	3.329	4.215
SGS-SLAM [19]	RMSE↓	0.463	0.216	0.300	0.339	0.547	0.299	0.451	0.311	0.365	-	-	-	-
GS-ICP SLAM [11]	RMSE↓	**0.189**	**0.132**	0.216	**0.201**	0.236	0.160	0.162	**0.117**	**0.177**	3.539	2.251	2.972	2.921
Ours	RMSE↓	0.296	0.264	**0.182**	0.429	0.354	1.040	0.434	0.441	0.430	3.187	0.370	1.026	1.528

Table 11. Camera tracking result on Replica and TUM RGB-D datasets for **RGB-D** camera. **RMSE of ATE** (cm) is reported. The best results are marked as **best score**, second best score and third best score. '-' denotes the system does not provide valid results.

Datasets (Camera)		Replica (**Mono**)									TUM RGB-D (**Mono**)			
Method	Metric	R0	R1	R2	Of0	Of1	Of2	Of3	Of4	Avg.	fr1/d	fr2/x	fr3/o	Avg.
ORB-SLAM3 [3]	RMSE↓	51.388	26.384	4.330	110.212	103.948	65.359	51.145	1.188	51.744	4.3269	10.4598	123.226	46.004
DRIOD-SLAM [37]	RMSE↓	103.892	53.146	66.939	53.267	34.431	119.311	98.089	83.732	76.600	1.769	0.458	2.839	1.689
MonoGS [26]	RMSE↓	12.623	56.357	25.350	43.245	19.729	39.148	11.754	88.230	37.054	4.575	4.605	2.847	4.009
Photo-SLAM [14]	RMSE↓	0.336	0.551	0.234	2.703	0.505	**2.065**	0.399	0.644	0.930	**1.633**	0.935	2.050	1.539
Ours	RMSE↓	**0.288**	**0.388**	**0.215**	**0.579**	**0.320**	3.963	**0.307**	**0.603**	**0.833**	3.187	**0.370**	**1.026**	**1.505**

Table 12. Camera tracking result on Replica and TUM RGB-D datasets for **monocular** camera. **RMSE of ATE** (cm) is reported. Best results are marked as **best score**, second best score and third best score.

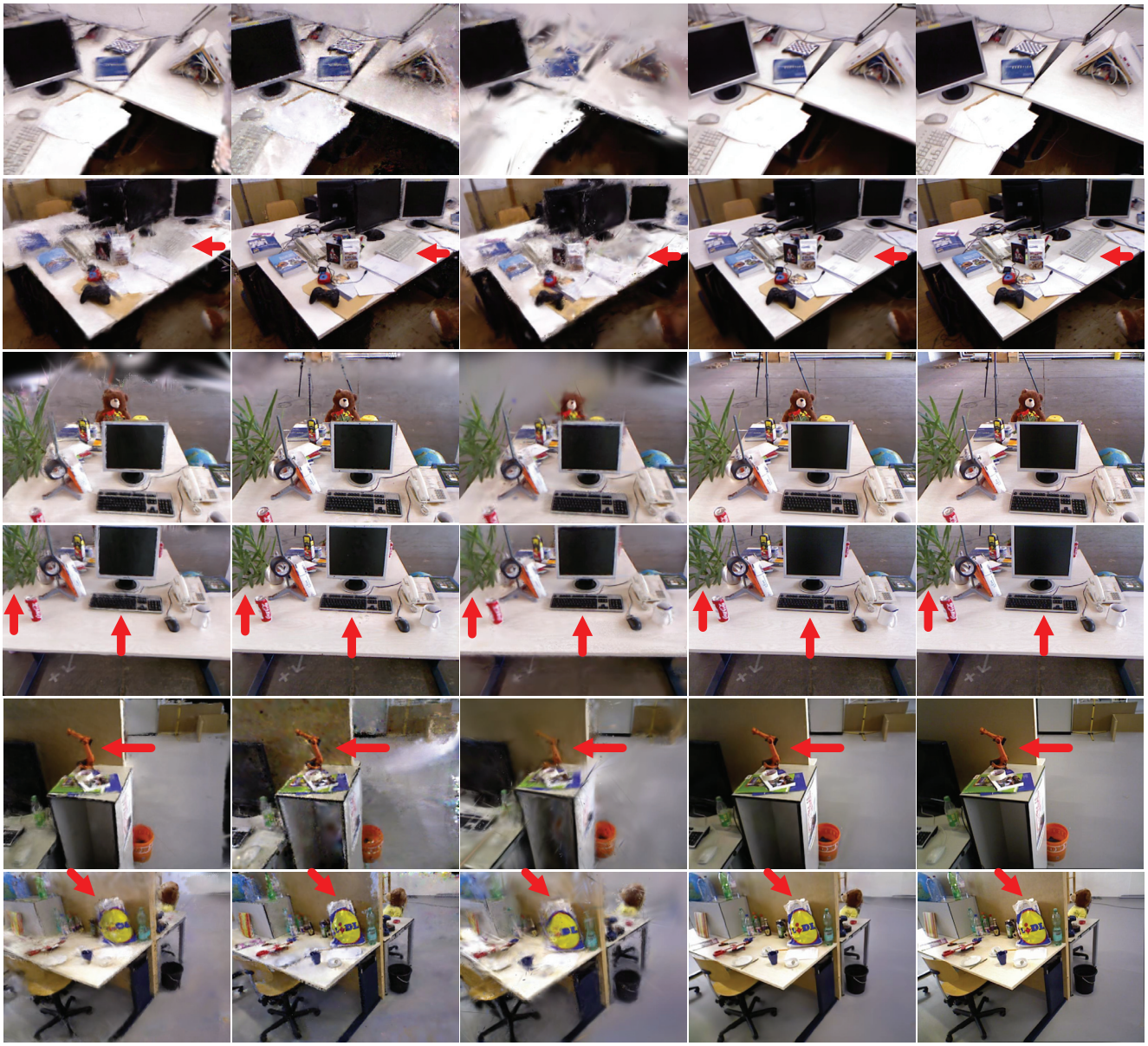
Datasets (Camera)		EuRoC (**Stereo**)				
Method	Metric	MH01	MH02	V101	V201	Avg.
MonoGS [26]	PSNR↑	**22.84**	**25.53**	23.39	18.66	22.60
	SSIM↑	**0.789**	**0.850**	0.831	0.687	0.789
	LPIPS↓	0.243	**0.181**	0.287	0.384	0.274
Photo-SLAM [14]	PSNR↑	11.22	11.14	13.78	11.46	11.90
	SSIM↑	0.300	0.306	0.520	0.509	0.409
	LPIPS↓	0.469	0.464	0.394	0.427	0.439
Photo-SLAM-30K	PSNR↑	11.10	11.04	13.66	11.26	11.77
	SSIM↑	0.296	0.300	0.516	0.508	0.405
	LPIPS↓	0.466	0.457	0.389	0.409	0.430
Ours	PSNR↑	22.50	22.30	**24.90**	**24.89**	**23.64**
	SSIM↑	0.750	0.727	**0.843**	**0.842**	**0.791**
	LPIPS↓	**0.220**	0.269	**0.122**	**0.117**	**0.182**

(a) Quantitative evaluation of our method compared to state-of-the-art methods for **Stereo** camera on EuRoC MAV datasets. The best results are marked as **best score** and second best score.

Datasets (Camera)		EuRoC (**Stereo**)				
Method	Metric	MH01	MH02	V101	V201	Avg.
ORB-SLAM3 [3]	RMSE↓	4.806	4.938	8.829	25.057	10.907
DRIOD-SLAM [37]	RMSE↓	**1.177**	**1.169**	**3.678**	**1.680**	**1.926**
MonoGS [26]	RMSE↓	11.194	8.327	29.365	148.080	49.241
Photo-SLAM [14]	RMSE↓	3.997	4.547	8.882	26.665	11.023
Ours	RMSE↓	3.948	3.863	8.823	13.217	7.462

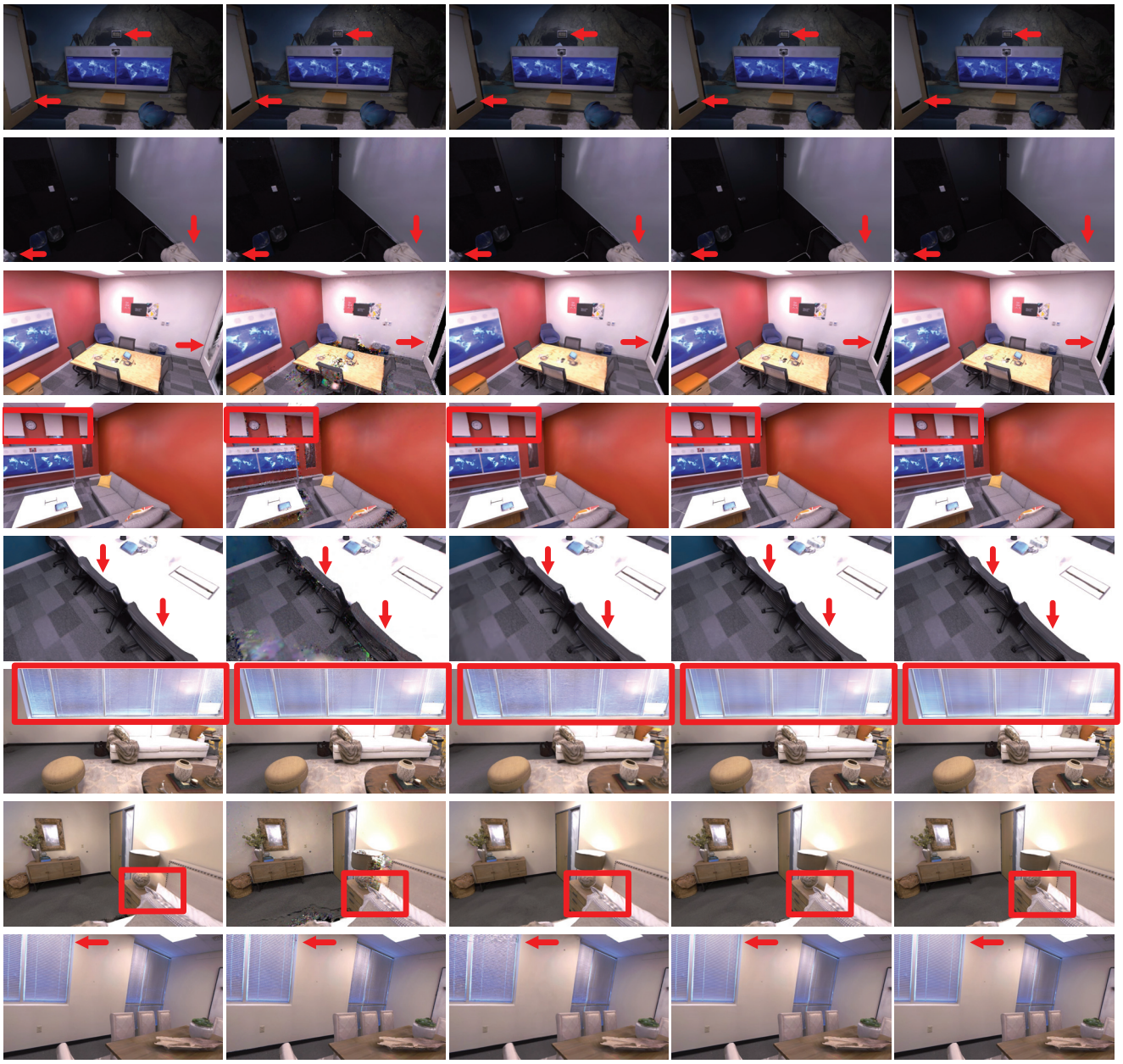
(b) Camera tracking result on EuRoC MAV datasets for **stereo** camera. **RMSE of ATE** (cm) is reported. The best results are marked as **best score** and second best score.

Table 13. Quantitative evaluation of our method compared to state-of-the-art methods for **Stereo** camera on EuRoC MAV datasets.



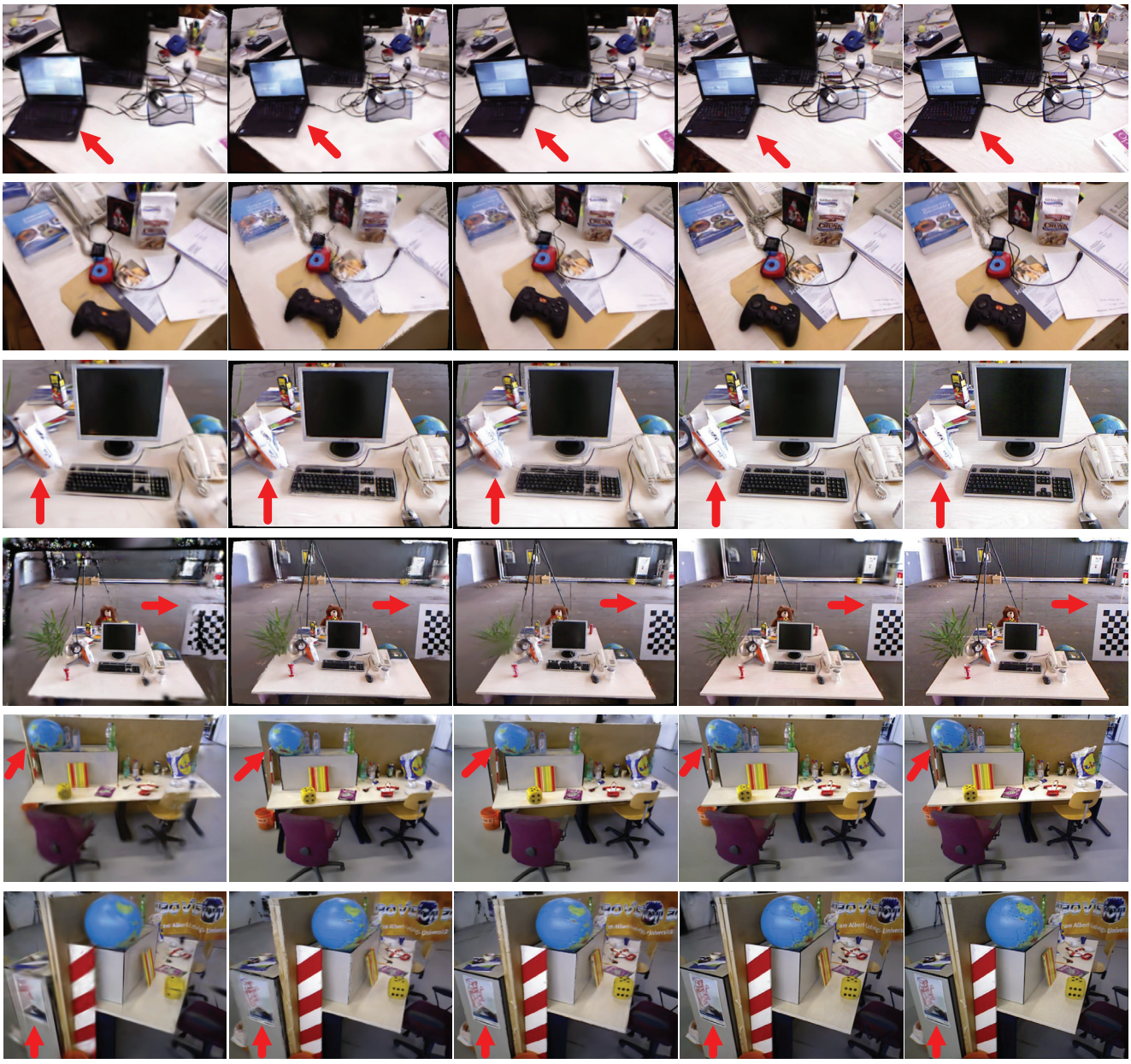
(a) GS-ICP SLAM [11] (b) SplaTAM [16] (c) RTG-SLAM [29] (d) **Ours** (e) Ground Truth

Figure 13. We show comparisons of ours to state-of-the-art methods on TUM RGB-D dataset for **RGB-D** camera. From top to bottom, the scenes are *fr1/desk* (rows 1–2), *fr2/xyz* (rows 3–4), and *fr3/office* (rows 5–6). Non-obvious differences in quality are highlighted by arrows.



(a) GS-ICP SLAM [11] (b) SplaTAM [16] (c) RTG-SLAM [29] (d) **Ours** (e) Ground Truth

Figure 14. We show comparisons of ours to state-of-the-art methods on Replica dataset for **RGB-D** camera. From top to bottom, the scenes are *Office0*, *Office1*, *Office2*, *Office3*, *Office4*, *room0*, *room1*, and *room2*. Non-obvious differences in quality are highlighted by arrows/insets. Since all methods achieve high-quality rendering results on the Replica dataset, we have highlighted specific regions in the images. In these annotated areas, our method consistently demonstrates sharper edges or finer textures.

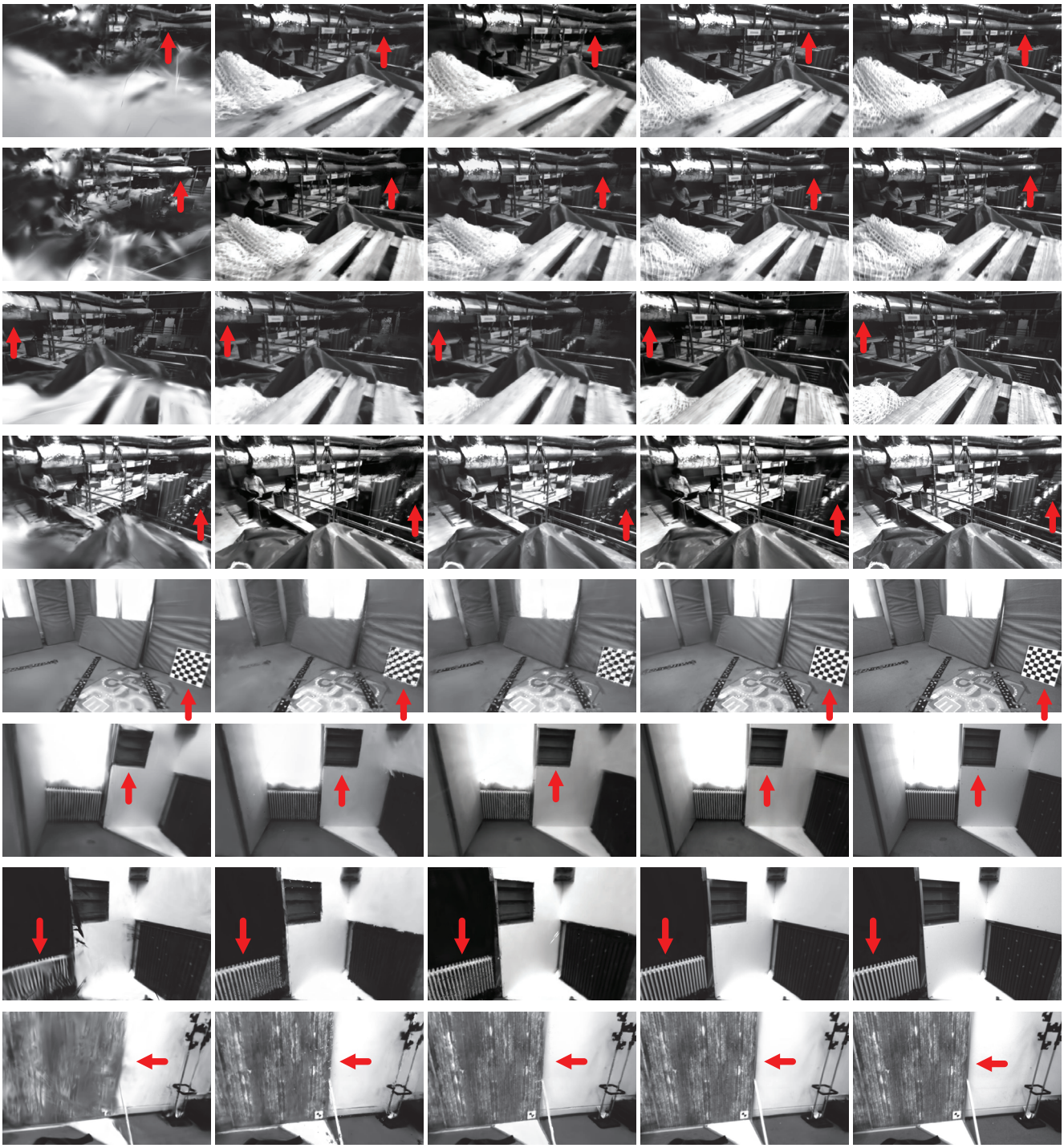


(a) MonoGS [26] (b) Photo-SLAM [14] (c) Photo-SLAM-30K (d) **Ours** (e) Ground Truth

Figure 15. We show comparisons of ours to state-of-the-art methods on TUM RGB-D dataset for **Monocular** camera. From top to bottom, the scenes are *fr1/desk* (rows 1–2), *fr2/xyz* (rows 3–4), and *fr3/office* (rows 5–6). Non-obvious differences in qualityare highlighted by arrows. Photo-SLAM [14] uses a set of parameters to undistort images as ground truth supervision. Consequently, its rendered images for *fr1/desk* and *fr2/xyz* exhibit black borders.

(a) MonoGS [26] (b) Photo-SLAM [14] (c) Photo-SLAM-30K (d) **Ours** (e) Ground Truth

Figure 16. We show comparisons of ours to state-of-the-art methods on Replica dataset for **Monocular** camera. From top to bottom, the scenes are *Office0*, *Office1*, *Office2*, *Office3*, *Office4*, *room0*, *room1*, and *room2*. Non-obvious differences in quality are highlighted by arrows/insets. Since all methods achieve high-quality rendering results on the Replica dataset, we have highlighted specific regions in the images. In these annotated areas, our method consistently demonstrates sharper edges or finer textures.



(a) MonoGS [26] (b) Photo-SLAM [14] (c) Photo-SLAM-30K (d) **Ours** (e) Ground Truth

Figure 17. We show comparisons of ours to state-of-the-art methods on the EuRoC MAV dataset for **Stereo** camera. From top to bottom, the scenes are *MH01* (rows 1–2), *MH02* (rows 3–4), *V101* (rows 5–6), and *V201* (rows 7–8). Non-obvious differences in quality are highlighted by arrows.