Splat-based 3D Scene Reconstruction with Extreme Motion-blur

Hyeonjoong Jang[†*] Dongyoung Choi[†] Donggun Kim[†] Woohyun Kang[†] Min H. Kim[†*]
[†] KAIST [*] HYPERGRAM
{hjjang; dychoi; dgkim; whkang; minhkim}@vclab.kaist.ac.kr

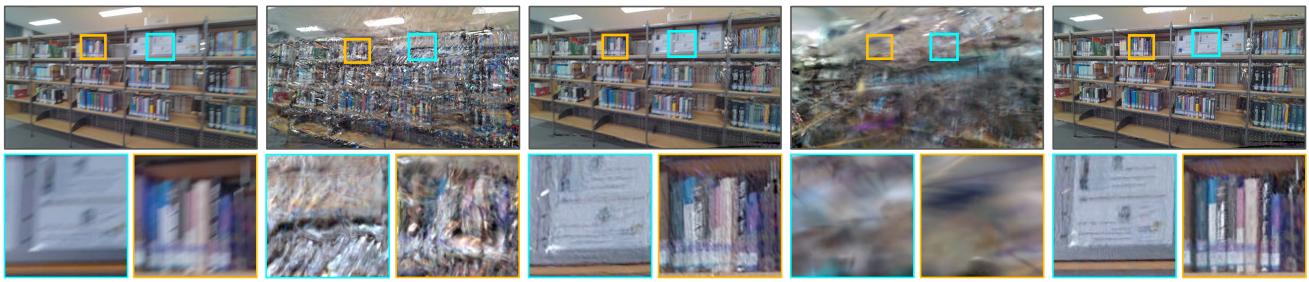

Figure 1. We propose a robust 3D scene reconstruction method from RGB-D input that effectively addresses extreme motion blur. Our approach achieves accurate camera poses and dense point clouds, producing clearer, deblurred scenes compared to existing methods.

Abstract

We propose a splat-based 3D scene reconstruction method from RGB-D input that effectively handles extreme motion blur, a frequent challenge in low-light environments. Under dim illumination, RGB frames often suffer from severe motion blur due to extended exposure times, causing traditional camera pose estimation methods, such as COLMAP, to fail. This results in inaccurate camera pose and blurry color input, compromising the quality of 3D reconstructions. Although recent 3D reconstruction techniques like Neural Radiance Fields and Gaussian Splatting have demonstrated impressive results, they rely on accurate camera trajectory estimation, which becomes challenging under fast motion or poor lighting conditions. Furthermore, rapid camera movement and the limited field of view of depth sensors reduce point cloud overlap, limiting the effectiveness of pose estimation with the ICP algorithm. To address these issues, we introduce a method that combines camera pose estimation and image deblurring using a Gaussian Splatting framework, leveraging both 3D Gaussian splats and depth inputs for enhanced scene representation. Our method first aligns consecutive RGB-D frames through optical flow and ICP, then refines camera poses and 3D geometry by adjusting Gaussian positions for optimal depth alignment. To handle motion blur, we model camera movement during exposure and deblur images by comparing the input with a series of sharp, rendered frames. Experiments on a new RGB-D dataset with extreme motion blur show that our method outperforms existing approaches, enabling high-quality reconstructions even in challenging conditions. This approach has broad implications for 3D mapping applications in robotics, autonomous navigation, and augmented reality. Both code and dataset are publicly available on https://github.com/KAIST-VCLAB/gs-extreme-motion-blur.

1. Introduction

High-quality 3D scene reconstruction is one of the most important and challenging applications in computer vision. The accuracy of 3D reconstruction hinges on the quality of essential components, such as camera poses, RGB images, and depth maps. These elements are interconnected and influence each other; a failure in one component can adversely impact the others. For example, blurry RGB images or noisy depth maps can significantly impair camera pose estimation, which, in turn, degrades overall reconstruction quality. This challenge is particularly pronounced when input data is captured in low-light conditions or during rapid camera movement—common scenarios in everyday, casual capture. Such conditions result in degraded color and depth frames, leading to poorly estimated camera poses and reconstructions with blurry textures and distorted or smoothed geometry. Therefore, achieving high-quality 3D reconstruction requires clear RGB images and accurate depth maps.

The interdependence of these input components suggests an opportunity for compensating or restoring degraded elements. However, this process presents a chicken-and-egg

problem, complicating the reconstruction further. For instance, accurate camera pose estimation requires sharp images, while deblurring motion-blurred images depend on reliable camera pose information. Although numerous methods [4, 41] exist for recovering sharp images from motion-blurred ones, they often rely on supervised learning models trained on specific datasets. Consequently, their deblurring performance can degrade significantly when faced with new cameras or motion blur scenarios that differ from the training data.

Recent works [35, 44, 45, 49] have made advances in addressing these challenges by optimizing camera trajectories during exposure while simultaneously learning sharp RGB colors. This enables simultaneous camera pose estimation and image deblurring. However, these methods require an initial camera pose and, when using Gaussian Splatting [14], also need a sparse point cloud as input. This requirement limits their applicability, especially when severe motion blur prevents structure-from-motion (SfM) methods like COLMAP [31] from functioning effectively, making it impossible to start the optimization process.

To overcome these limitations, we introduce a robust approach for 3D scene reconstruction that effectively compensates for severe motion blur and addresses the challenges of camera pose estimation without relying on a precise initial pose. Our method leverages RGB-D inputs and incorporates both optical flow and depth information to align camera poses accurately, even in the presence of challenging motion blur and lighting conditions. We first perform a global alignment between consecutive frames using optical flow and the ICP algorithm, which enables effective local alignment of point clouds generated from the depth maps.

After this initial alignment, we refine the camera poses and 3D geometry by integrating them into a Gaussian Splatting pipeline. This approach allows us to initialize dense 3D Gaussians from depth maps, which we scale to ensure a detailed representation of the scene geometry. Our refinement process iteratively adjusts both camera poses and the positions of 3D Gaussians, achieving global consistency by minimizing a depth alignment loss that compares rendered depth maps with input depth measurements. Through this adjustment, we eliminate loop-closure artifacts and reduce the drift that often accumulates in traditional SLAM-based methods.

For scenes with significant motion blur, we further optimize the deblurring process by modeling the camera poses at the start and end of each frame's exposure. This temporal modeling allows us to simulate the effects of motion during the exposure period, which we incorporate into our Gaussian Splatting framework. We minimize the difference between the observed motion-blurred image and the average of a set of sharp images rendered from multiple viewpoints along the exposure trajectory, resulting in more accurate, geometrically consistent deblurring. Our deblur loss function combines image alignment, structural similarity, and depth consistency, ensuring that the final reconstructions retain sharpness and fidelity to the original scene structure.

Through extensive evaluation on a newly constructed RGB-D dataset featuring extreme motion blur, we demonstrate that our approach significantly outperforms existing methods in both accuracy and robustness. By addressing the key challenges of pose estimation, depth alignment, and image deblurring, our method provides a versatile and effective solution for high-quality 3D reconstruction under real-world capture conditions, such as low light and fast motion. The proposed method holds promising applications in areas requiring reliable 3D mapping and reconstruction, including robotics, autonomous navigation, and augmented reality. We will make both our dataset and implementation publicly available to foster further research and development in this field.

2. Related Work

RGB-D scene reconstruction. A popular approach to 3D geometry reconstruction defines a 3D voxel grid and constructs a signed distance function (SDF) volume by accumulating depth estimates captured by sensors. KinectFusion [27] uses a Kinect camera to capture RGB-D images, then estimates relative camera poses between adjacent frames with the ICP algorithm. Next, it constructs and updates a truncated signed distance function (TSDF) volume, enabling real-time 3D reconstruction of objects. Niessner et al. [28] improve memory efficiency by combining hashing algorithms to store and update SDF values only where needed, thus reducing memory requirements.

Several methods [8, 11, 20, 22] adopt memory-efficient, hierarchical data structures for depth fusion. BundleFusion [6] simultaneously refines camera poses and geometry using bundle adjustment, improving reconstruction accuracy and addressing the loop closure problem caused by accumulated camera pose and depth estimation errors. ElasticFusion [47] builds a map with surfels—point primitives containing position, normal, and radius—and updates the map through a deformation graph to minimize error and perform loop closure. Other methods [13, 33, 38] directly fuse depth estimates into point clouds, optimizing memory use. However, these methods generally require slow camera motion to capture sharp images and accurately estimate camera transformations.

Simultaneous localization and mapping (SLAM) methods [3, 15, 25, 32, 46] track camera motion and construct a map in real-time by extracting visual features from input images to establish 3D point correspondences. CodeSLAM [2] and DeepTAM [50] use neural networks to combine depth maps with color images, while Azinović

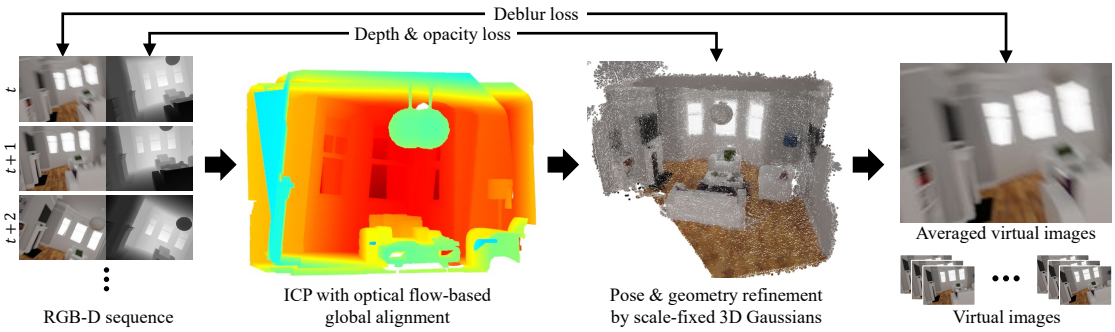

Figure 2. Overview of our method pipeline. We begin with global alignment by comparing optical flow and reprojected pixel differences between adjacent RGB-D image pairs, followed by local estimation of relative camera poses using ICP. Each depth map and camera pose is then used to initialize 3D Gaussians. By fixing the scale of the Gaussians and optimizing with depth and opacity losses, we achieve global refinement of camera poses and geometry. Finally, we deblur the scene texture by minimizing the deblur loss between the input image and averaged virtual images.

et al. [1] train multi-layer perceptrons (MLP) to learn SDF and color values for scene reconstruction. Their approach also optimizes camera transformation correction variables during training to refine poses. However, these methods struggle with input images that contain motion blur. Nice-SLAM [51], Point-SLAM [30], and DROID-SLAM [42] use neural networks to track camera poses. SplaTAM [12] and MonoGS [21] introduced SLAM solutions based on 3D Gaussian Splatting. However, these methods are not designed to handle motion blur.

Image deblurring. Motion blur occurs when camera motion during the exposure time changes the projected pixel coordinates of rays on the camera sensor. Early studies [7, 16, 36] tackle this issue by developing kernels to restore blurry images. Convolutional neural networks (CNNs) demonstrate powerful performance in learning deblur kernels [34, 40], and deep learning techniques such as ResNet with skip connections and multi-scale networks further improved restoration quality [26, 39]. SRN-DeblurNet [41] adopt recurrent networks [10, 37] and combine them with an encoder-decoder network structure, while NAFNet [4] simplify the network structure, extracting only essential components and proposing a nonlinear activation-free network. However, since these methods rely on curated datasets like the GoPro dataset [26], they struggle to effectively deblur images with extreme motion blur or those captured on different cameras.

There are NeRF-based approaches [24] that tackle motion blur in input images. Deblur-NeRF [19] and DP-NeRF [17] use a set of motion-blurred images along with camera poses estimated by COLMAP as input to restore sharp images through NeRF optimization. BAD-NeRF [44] enhances this by jointly optimizing virtual camera poses and radiance fields during exposure. The latest approaches utilize Gaussian Splatting [14], which offers faster training times than NeRF by rendering a set of 3D Gaussians with a dedicated rasterizer rather than optimizing a neural network. Gaussian Splatting-based deblurring methods [35, 45, 49] optimize the virtual camera trajectory and minimize differences between input blurry images and the average of rendered images at each virtual camera position.

All of these methods rely on initial camera poses, typically obtained from COLMAP, and for Gaussian Splatting-based methods, a sparse point cloud as well. However, structure-from-motion methods like COLMAP often fail with blurry inputs—such as images captured by fast-moving cameras or in low-light conditions that require long exposure times. Even with dense depth inputs, ICP requires acceptable global alignment, and recent point cloud matching methods [9, 18] cannot accurately determine relative camera transformations between two point clouds if they consist mainly of planar structures.

To address these issues, we propose an effective camera pose estimation algorithm tailored for RGB-D image sequences that jointly refines both the camera trajectory and the reconstructed 3D point cloud, resulting in improved sharpness in 3D reconstruction.

3. Method

Overview. In datasets with significant motion blur, using COLMAP to estimate camera poses becomes impractical. Additionally, rapid camera motion and the limited field of view of depth sensors further complicate this task, making ICP registration less effective. To address these challenges, we first perform global alignment using both RGB and depth images, which establishes sufficient initial alignment for the ICP algorithm to operate effectively. After achieving this global alignment, we apply ICP again to refine and accurately estimate the camera poses. However, estimating camera poses by analyzing only two consecutive frames in a sequence can result in accumulated errors, leading to drift over time. To resolve this issue, it is crucial to adjust the 3D geometry and align the camera poses across the entire sequence so that depth maps and camera views from all perspectives remain well-aligned. We accomplish this through bundle adjustment, optimizing the positions of

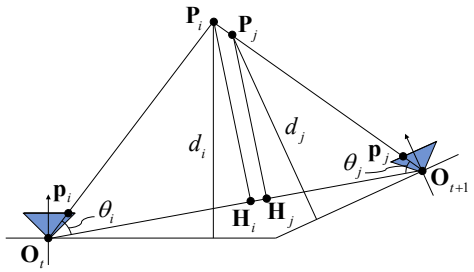

Figure 3. The illustration of our pose initialization loss. When $\mathbf{p}_i$ is reprojected to $\mathbf{p}_j$ using the depth value, it is considered geometrically consistent if $\overline{\mathbf{P}_i\mathbf{H}_i}$ equals $\overline{\mathbf{P}_j\mathbf{H}_j}$.

3D Gaussians and camera poses within the Gaussian Splatting pipeline. Finally, to account for motion blur, we approximate the camera trajectory over time and optimize the movement during exposure, enabling effective deblurring. See Figure 2 for an overview.

3.1. Local Pose Estimation

To estimate relative camera transformations, we select consecutive frames from the input RGB-D image sequence. Our approach begins with global alignment using optical flow, followed by local alignment of the two-point clouds using the ICP algorithm. For each consecutive RGB image and depth map pair at times t and $t+1$, we optimize the transformation $\boldsymbol{\xi} \in \mathfrak{se}(3)$ between the frames. Here, $\boldsymbol{\xi}_{t\to t+1} = [\mathbf{x}^\intercal, \boldsymbol{\omega}^\intercal]^\intercal$, where $\mathbf{x}$ represents the translation vector between the two camera origins, and $\boldsymbol{\omega}$ is the rotation element of $\mathfrak{se}(3)$. We ultimately calculate a camera-to-world transformation matrix $\boldsymbol{\xi}_t$ for each timestamp t.

To achieve this, we first compute the difference between the estimated optical flow and the reprojected pixel from the camera at time t to $t+1$ using the depth estimate for each pixel in the depth map D_t and the intrinsic matrix $\mathbf{K}$. A pixel $\mathbf{p}_i$ in the depth map is backprojected to 3D space as: $\mathbf{P}_i = \pi^{-1}(\mathbf{K}, \mathbf{p}_i, D_t(\mathbf{p}_i))$. We then project this 3D point to the camera at $t+1$: $\mathbf{p_j} = \pi(\mathbf{K}, \exp(\boldsymbol{\xi}^\wedge_{t\to t+1})\mathbf{P}_i)$, where π and π^{-1} represent the projection and backprojection operators, respectively. The pixel movement from $\mathbf{p}_i$ to $\mathbf{p}_j$ is then compared with the estimated optical flow $F_{t\to t+1}(\mathbf{p}_i)$ to compute the optical flow loss:

$$\mathcal{L}_F = \sum_i M_t(\mathbf{p}_i) M_{t+1}(\mathbf{p}_j) \| F_{t\to t+1}(\mathbf{p}_i) - (\mathbf{p}_j - \mathbf{p}_i) \|_2.$$

In this formulation, M_t is a mask indicating valid depth values in D_t at pixel $\mathbf{p}_i$. The mask M_t dynamically updates during optimization, as $\mathbf{p}_j$ depends on the transformation $\boldsymbol{\xi}$: $M_t(\mathbf{p}_i) = D_t(\mathbf{p}_i) > 0$.

We employ a state-of-the-art pretrained optical flow estimation model [48] designed hierarchically to produce reliable optical flow even from two blurry input images. Next, we calculate the photometric loss between color values at pixel $\mathbf{p}_i$ in image I_t and pixel $\mathbf{p}_j$ in image I_{t+1}:

$$\mathcal{L}_I = \sum_i M_t(\mathbf{p}_i) M_{t+1}(\mathbf{p}_j) \| I_t(\mathbf{p}_i) - I_{t+1}(\mathbf{p}_j) \|_1.$$

To ensure geometric consistency, we introduce a depth consistency loss by using the two depth maps captured by the depth camera to optimize relative poses. Specifically, as illustrated in Figure 3, we compute and compare the vertical components $\overline{\mathbf{P}_i\mathbf{H}_i}$ and $\overline{\mathbf{P}_j\mathbf{H}_j}$ based on the baseline vector $\overline{\mathbf{O}_t\mathbf{O}_{t+1}}$ between the two cameras. We calculate $\overline{\mathbf{P}_i\mathbf{H}_i}$ as follows:

$$\overline{\mathbf{P}_i\mathbf{H}_i} = \|\mathbf{P}_i\|_2 \sin\left(\cos^{-1}\left(\frac{\mathbf{P}_i \cdot \mathbf{x}}{\|\mathbf{P}_i\|_2 \|\mathbf{x}\|_2}\right)\right).$$

Similarly, we compute $\overline{\mathbf{P}_j\mathbf{H}_j}$ as:

$$\overline{\mathbf{P}_j\mathbf{H}_j} = \|\mathbf{P}_j\|_2 \sin\left(\cos^{-1}\left(\frac{\mathbf{P}_j \cdot (-\exp(\boldsymbol{\omega})^\intercal \mathbf{x})}{\|\mathbf{P}_j\|_2 \| -\exp(\boldsymbol{\omega})^\intercal \mathbf{x}\|_2}\right)\right).$$

This depth consistency loss is calculated as follows:

$$\mathcal{L}_C = \sum_i M_t(\mathbf{p}_i) M_{t+1}(\mathbf{p}_j) \left\| \overline{\mathbf{P}_i\mathbf{H}_i} - \overline{\mathbf{P}_j\mathbf{H}_j} \right\|, \quad (1)$$

where exp() is the exponential mapping operator in Lie algebra, which helps maintain alignment. Our final loss function combines these terms with weighted coefficients:

$$\mathcal{L}_{\text{pose}} = \mathcal{L}_F + \lambda_I \mathcal{L}_I + \lambda_C \mathcal{L}_C. \quad (2)$$

We start optimization with $\lambda_I = 0$ and $\lambda_C = 0$, allowing only the flow loss $\mathcal{L}_F$ to guide the relative pose alignment with global optical flow. After initial convergence, we increase weights to $\lambda_I = 75$ and $\lambda_C = 25$ and continue optimization to a second convergence.

Finally, we apply a point-to-plane ICP algorithm on the two globally aligned point clouds to estimate camera poses accurately. Since optical flow estimation between two blurry images introduces some errors, this final camera pose estimation relies exclusively on depth maps.

3.2. Global Pose Estimation/Geometry Refinement

Starting from the initial camera poses and point clouds obtained in the previous step, we merge them into a global point cloud within the world coordinate system (Figure 4). To manage memory efficiently, we downsample the point cloud with a specified voxel size s. However, due to accumulated errors in the initial camera poses, these poses may not align well with the global point cloud, leading to seam artifacts at image boundaries and a drifting phenomenon that creates loop-closure issues. To resolve this, we refine both the camera poses and 3D point cloud positions simultaneously within the Gaussian Splatting pipeline, ensuring global and geometric consistency.

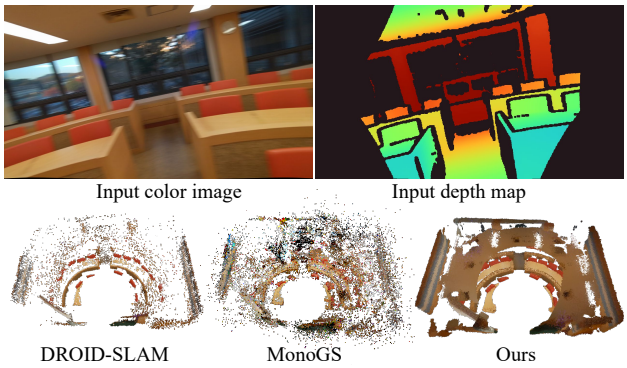

Figure 4. Reconstructed initial point clouds from the RGB-D image sequence. Our method achieves accurate camera poses and generates a dense, high-quality point cloud.

We initialize the Gaussian Splatting pipeline by loading the point cloud along with the estimated camera poses. To capture every depth value per pixel from the dense depth maps, we initialize 3D Gaussians, fixing their sizes and setting their scale to $s/2$. This dense arrangement of 3D Gaussians provides a detailed geometric representation. We then train only the camera poses, 3D positions, and opacities of the Gaussians by comparing a rendered depth map $\widetilde{D}$ with the input depth map D.

During training, we randomly select camera viewpoints from among the trainable camera pose variables to render each depth map. For a depth value $\widetilde{D}_t(\mathbf{p}_i)$ at pixel $\mathbf{p}_i$ in the rendered map, we calculate the depth loss as:

$$\mathcal{L}_D = \sum_i M_t(\mathbf{p}_i) \left\| D_t(\mathbf{p}_i) - \widetilde{D}_t(\mathbf{p}_i) \right\|_1. \tag{3}$$

We also initialize each 3D Gaussian's opacity o_j to 0.5, optimizing them to values of 0 or 1:

$$\mathcal{L}_{\text{opacity}} = \sum_j o_j^2 (1 - o_j)^2. \tag{4}$$

This optimization aims to retain only Gaussians with an opacity of 1, pruning others to reduce ambiguity for the subsequent deblurring stage. We prune Gaussians with an opacity less than 0.8 and reset opacity with 0.5 again every certain amount of iteration. As iterations progress, the rendered depth maps increasingly align with the input depth maps, indicating that all 3D Gaussian positions and camera poses are globally well-aligned (Figure 5).

The final refinement loss is a weighted combination of the depth and opacity losses:

$$\mathcal{L}_{\text{refinement}} = \lambda_D \mathcal{L}_D + \lambda_o \mathcal{L}_{\text{opacity}}. \tag{5}$$

3.3. Image Deblur

Motion blur caused by camera movement can generally be approximated as:

$$I(t) \approx B(t) = \frac{1}{k} \sum_{i=0}^{k} C_i(t) \tag{6}$$

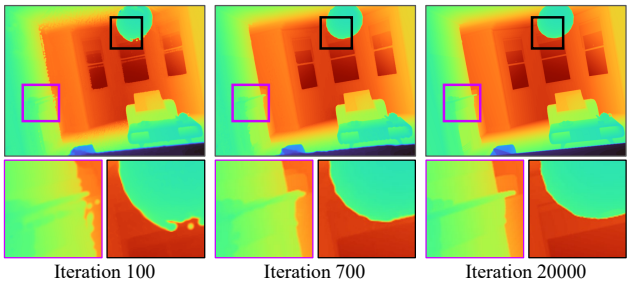

Figure 5. The refinement process corrects misaligned 3D Gaussians by updating camera poses and geometry, with effectiveness demonstrated in Table 3.

Here, $B(t)$ represents the motion-blurred image, which we model as the average of k sharp images $C_i(t)$ rendered from the 3D Gaussians at k virtual camera poses interpolated between the camera poses at the start and end of the exposure time period $\boldsymbol{\xi}_t^s$ and $\boldsymbol{\xi}_t^e$. This approximation allows $B(t)$ to closely match the input image $I(t)$.

Using the refined camera poses and point cloud from the previous step, we incorporate depth information to achieve geometrically accurate deblurring, following the approach in recent work [49]. Specifically, we minimize the difference between the input image $I(t)$ and the averaged rendered image $B(t)$, as shown below:

$$\mathcal{L}_B = \sum_t \sum_i \| I_t(\mathbf{p}_i) - B_t(\mathbf{p}_i) \|_1. \tag{7}$$

The final deblur loss function integrates this alignment loss with additional terms for structural similarity and depth consistency:

$$\mathcal{L}_{\text{deblur}} = (1 - \lambda_B)\mathcal{L}_B + \lambda_B \mathcal{L}_{\text{D-SSIM}} + \lambda_D \mathcal{L}_D. \tag{8}$$

This combined loss helps ensure that the deblurred output aligns well with both the input image and depth structure, leading to sharper and more geometrically consistent results.

4. Results

Dataset preparation. We prepare both synthetic and real-world datasets to evaluate our method's performance. In the absence of an RGB-D dataset with extreme motion blur, we generate a synthetic reference dataset by rendering directly from 3D models. We create RGB-D image sequences and record per-frame camera trajectories to evaluate our pose estimation and deblurring accuracy. The sequences, ranging from 50 to 150 frames, are captured along simulated trajectories that include rapid translations, 360-degree rotations, and human movement paths (using Blender's Walk Navigation feature). The RGB and depth images have a resolution of 640×480, and motion blur is applied by enabling Blender's motion blur setting. The 3D models are sourced from the McGuire archive [23] and the Blender archive [43].

Figure 6. Qualitative comparison of deblurring performance from a novel view. Our method excels in restoring high-frequency details, whereas several methods, including COLMAP [31], fail under severe motion blur conditions (see Section 4.1). The corresponding quantitative results are provided in Table 2.

For real scene evaluation, we used an Azure Kinect RGB-D camera with a 33.3ms fixed exposure time, 5000K fixed white balance, NFOV-2 $\times$ 2-binned depth capture mode, and 1280 $\times$ 720 RGB resolution. We used the provided intrinsic camera parameters to undistort the images and aligned depth maps to color images using the SDK's transformation functions. Depth maps were eroded three times to remove outliers, which commonly occur along edges.

Implementation details. Our code is based on 3D Gaussian Splatting [14], with data processing partially adapted from CF-3DGS [14]. For initial camera pose estimation, we use the Unimatch [48] optical flow network and the ICP [5] algorithm from Open3D as part of the optimization process. For an input image sequence of N frames, we prune Gaussians every $100N$ iterations, reset opacity to 0.5 every $200N$ iterations, and run a total of $600N$ iterations (Section 3.2). The number of virtual views for deblurring is set to 15 for synthetic dataset evaluation and 17 for real-scene comparison. All experiments were conducted on an NVIDIA RTX A6000 GPU (48GB), with initial camera pose optimization taking 11 seconds per frame, 0.1 seconds per iteration for pose and geometry refinement, and 0.26 seconds per iteration for deblurring at 640×480 resolution.

4.1. Quantitative Evaluation

We evaluate the camera pose accuracy and deblurring performance on three synthetic scenes from our dataset, each rendered with extreme motion blur. For pose accuracy, we compare our method against a variety of approaches capable of generating a 3D point cloud and camera trajectory:

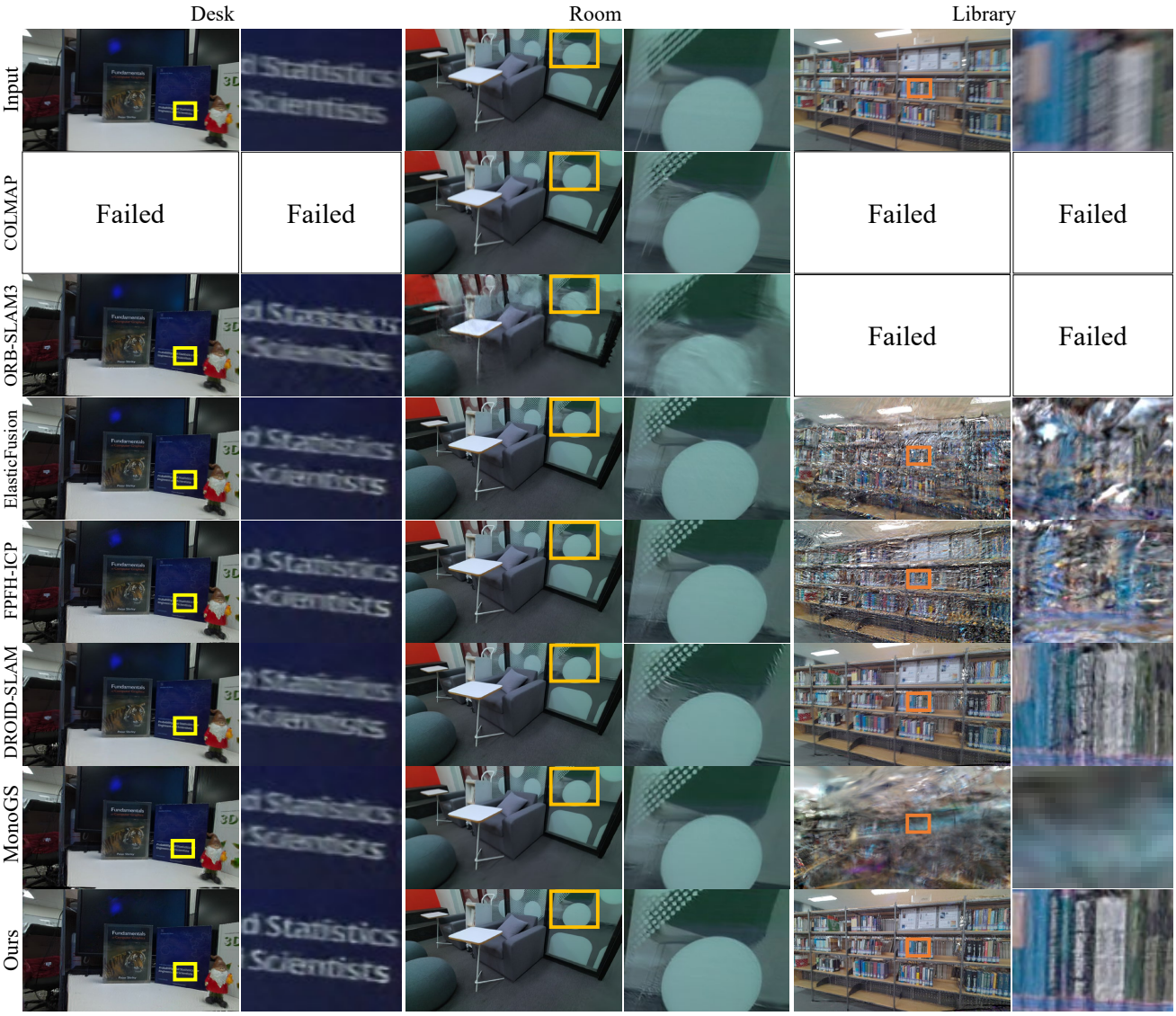

Figure 7. Qualitative comparison of deblurring performance from novel views on real scenes captured with an Azure Kinect camera. Our method effectively restores high-frequency details, outperforming other methods even when processing RGB-D sequences with large spatial gaps between frames due to rapid camera motion and severe motion blur.

COLMAP [31], ICP with FPFH feature-based global alignment [29], ORB-SLAM3 [3], ElasticFusion [47], and several RGB-D SLAM methods, including NICE-SLAM [51], Point-SLAM [30], and DROID-SLAM [42]. Additionally, we include recent Gaussian Splatting-based SLAM approaches such as MonoGS [21] and SplaTAM [12]. For each method, we calculate absolute trajectory error (ATE) and relative pose error (RPE) in meter and degree units, using the ground truth camera trajectories as references.

To evaluate deblurring accuracy, we modify a state-of-the-art Gaussian Splatting-based image restoration method [49] to operate in a depth-aware manner using undistorted color and depth images. We select one out of every three images as a test view and render the corresponding novel view for comparison with the ground truth. For quantitative metrics, we calculate PSNR, SSIM, and LPIPS on the rendered images, as well as RMSE between the rendered and ground truth depth maps in the inverse depth domain (m^{-1}).

4.2. Ablation Study

To validate the contributions of each component in our method, we perform an ablation study evaluating camera pose and deblurring accuracy by selectively omitting key elements. Specifically, we conduct experiments without fixing the scale of the Gaussians (Section 3.2), without the pose and geometry refinement process (Equation 5), and without depth loss (Equation 8).

Table 1. Quantitative comparison of camera pose accuracy against ground truth poses. We report scores only for methods that successfully provide both camera poses and a point cloud (see Section 4.1). Our method achieves the best scores on RPE metrics and shows ATE accuracy comparable to DROID-SLAM [42]. Each color highlights the best and second best results.

	Bedroom			Livingroom			Office		
	ATE ↓	RPE (trans) ↓	RPE (rot) ↓	ATE ↓	RPE (trans) ↓	RPE (rot) ↓	ATE ↓	RPE (trans) ↓	RPE (rot) ↓
ElasticFusion [47]	1.303	0.079	2.218	1.014	0.056	2.777	0.479	0.038	0.480
FPFH-ICP [29]	2.760	0.118	2.310	2.028	0.092	1.213	0.139	0.009	0.167
DROID-SLAM [42]	0.059	0.044	1.212	0.050	0.014	0.317	0.011	0.011	0.245
MonoGS [21]	0.506	0.041	0.812	0.135	0.009	0.124	0.045	0.011	0.180
Ours	0.102	0.006	0.092	0.005	0.002	0.032	0.041	0.003	0.046

Table 2. Quantitative comparison of deblurring performance from novel views. Our method demonstrates high accuracy across all metrics for both color and depth images. RMSE is calculated by comparing the rendered depth with GT depth in the inverse depth domain (m^{-1}).

	Bedroom				Livingroom				Office			
	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE ↓
ElasticFusion [47]	18.766	0.586	0.466	1.249	13.799	0.448	0.554	0.146	19.052	0.533	0.401	0.079
FPFH-ICP [29]	20.623	0.617	0.418	0.450	18.172	0.598	0.422	0.102	22.212	0.684	0.241	0.033
DROID-SLAM [42]	21.375	0.666	0.302	0.013	21.222	0.735	0.250	0.035	25.680	0.769	0.144	0.013
MonoGS [21]	23.728	0.717	0.285	0.009	24.414	0.822	0.188	0.027	24.194	0.725	0.184	0.184
Ours	26.745	0.824	0.206	0.005	27.650	0.900	0.150	0.010	25.649	0.791	0.160	0.014

Table 3. Ablation study on the effects of the global refinement process and fixed scale of Gaussians. Our refinement significantly enhances deblurring and depth accuracy while fixing the scale of the Gaussians leads to more accurate camera pose estimation.

	PSNR	SSIM	LPIPS	RMSE	ATE	RPE(trans)	RPE(rot)
w/o scale fix	21.656	0.703	0.308	0.027	0.155	0.053	1.488
w/o refinement	22.223	0.734	0.238	0.027	0.123	0.007	0.047
w/o depth loss	25.518	0.807	0.170	0.014	0.103	0.012	0.195
Scale fix + refinement	26.681	0.838	0.172	0.010	0.103	0.012	0.195

We assess the effectiveness of these components by performing deblurring based on results from Section 3.1—omitting refinement of camera poses and geometry—and by allowing the scale of the 3D Gaussians to be optimized freely. Table 3 demonstrates that both the global refinement process and fixed Gaussian scale are essential for achieving high accuracy. Notably, during pose refinement, the algorithm reduces absolute pose error even at the expense of a slight increase in relative pose error, underscoring the importance of our refinement approach for stable, high-quality reconstruction.

Our method achieves superior RPE scores across all scenes, as shown in Table 1, outperforming other approaches in pose and geometry estimation under severe motion blur. While our ATE scores are comparable to those of DROID-SLAM, which uses a bundle adjustment (BA) module, ATE alone does not correlate strongly with deblurring quality. Our method consistently produces more accurate and denser point clouds, which directly contributes to improved deblurring performance, as evidenced by the metrics in Table 2. These results confirm that our integrated approach to pose estimation and deblurring is more resilient to challenging conditions than other methods.

4.3. Qualitative Evaluation

Figures 6 and 7 present qualitative comparisons of our method's performance in synthetic and real scenes. The densification capability of Gaussian Splatting provides detailed scene representation even from a relatively sparse point cloud; however, our approach, with its initially dense and accurate point cloud generation, achieves notably superior deblurring performance, especially in restoring high-frequency details. Our method consistently demonstrates a significant advantage in visual fidelity and detail restoration across diverse conditions.

5. Conclusion

We have introduced a robust method for 3D scene reconstruction from RGB-D image sequences that effectively addresses the challenges of extreme motion blur and low-light conditions. Our approach leverages optical flow from color images and a carefully designed geometric loss to achieve accurate global alignment, followed by local refinement of camera poses and geometry using ICP. By integrating these steps into a Gaussian Splatting pipeline, we further refine camera poses and 3D geometry by minimizing depth and opacity losses. Additionally, fixing the scale of the 3D Gaussians ensures that the depth information from the RGB-D input is fully utilized, allowing for precise camera pose estimation and improved deblurring performance. Our method demonstrates superior performance on real and synthetic RGB-D scenes with significant motion blur, outperforming existing approaches.

Acknowledgements

Min H. Kim acknowledges the Samsung Research Funding & Incubation Center (SRFC-IT2402-02), the Korea NRF grant (RS-2024-00357548), the MSIT/IITP of Korea (RS-2022-00155620, RS-2024-00398830, 2022-0-00058, and 2017-0-00072), Microsoft Research Asia, and Samsung Electronics.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6290–6301, 2022. 3

[2] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2560–2568, 2018. 2

[3] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 2, 7

[4] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*, 2022. 2, 3

[5] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 6

[6] Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *ACM Transactions on Graphics 2017 (TOG)*, 2017. 2

[7] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *Acm Siggraph 2006 Papers*, pages 787–794. 2006. 3

[8] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Deep structured implicit functions. *arXiv preprint arXiv:1912.06126*, 2:2, 2019. 2

[9] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5545–5554, 2019. 3

[10] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012. 3

[11] Olaf Kähler, Victor Prisacariu, Julien Valentin, and David Murray. Hierarchical voxel block hashing for efficient integration of depth images. *IEEE Robotics and Automation Letters*, 1(1):192–197, 2015. 2

[12] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024. 3, 7

[13] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 1–8. IEEE, 2013. 2

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3, 6

[15] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013. 2

[16] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. *Advances in neural information processing systems*, 22, 2009. 3

[17] Dogyoon Lee, Minhyeok Lee, Chajin Shin, and Sangyoun Lee. Dp-nerf: Deblurred neural radiance field with physical scene priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12386–12396, 2023. 3

[18] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5554–5564, 2022. 3

[19] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V Sander. Deblur-nerf: Neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12861–12870, 2022. 3

[20] Nico Marniok, Ole Johannsen, and Bastian Goldluecke. An efficient octree design for local variational range image fusion. In *Pattern Recognition: 39th German Conference, GCPR 2017, Basel, Switzerland, September 12–15, 2017, Proceedings 39*, pages 401–412. Springer, 2017. 2

[21] 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. 3, 7, 8

[22] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pages 4628–4635. IEEE, 2017. 2

[23] Morgan McGuire. Computer graphics archive, 2017. 5

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[25] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. 2

[26] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 3

[27] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon.

Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 2

[28] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32 (6):1–11, 2013. 2

[29] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009. 7, 8

[30] 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. 3, 7

[31] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 6, 7

[32] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 134–144, 2019. 2

[33] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. Surfelmeshing: Online surfel-based mesh reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2494–2507, 2019. 2

[34] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1439–1451, 2015. 3

[35] Otto Seiskari, Jerry Ylilammi, Valtteri Kaatrasalo, Pekka Rantalankila, Matias Turkulainen, Juho Kannala, and Arno Solin. Gaussian splatting on the move: Blur and rolling shutter compensation for natural camera motion, 2024. 2, 3

[36] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. *Acm transactions on graphics (tog)*, 27(3):1–10, 2008. 3

[37] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 3

[38] Jörg Stückler and Sven Behnke. Multi-resolution surfel maps for efficient dense 3d modeling and tracking. *Journal of Visual Communication and Image Representation*, 25(1):137–147, 2014. 2

[39] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1279–1288, 2017. 3

[40] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 769–777, 2015. 3

[41] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[42] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*, 2021. 3, 7, 8

[43] Flavio Della Tommasa. Blender demo files, 2024. 5

[44] Peng Wang, Lingzhe Zhao, Ruijie Ma, and Peidong Liu. BAD-NeRF: Bundle Adjusted Deblur Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4170–4179, 2023. 2, 3

[45] Chen Wenbo and Liu Ligang. Deblur-gs: 3d gaussian splatting from camera motion blurred images. *Proc. ACM Comput. Graph. Interact. Tech. (Proceedings of I3D 2024)*, 7(1), 2024. 2, 3

[46] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J Leonard, and John McDonald. Real-time large-scale dense rgb-d slam with volumetric fusion. *The International Journal of Robotics Research*, 34(4-5):598–626, 2015. 2

[47] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016. 2, 7, 8

[48] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 4, 6

[49] Lingzhe Zhao, Peng Wang, and Peidong Liu. BAD-Gaussians: Bundle Adjusted Deblur Gaussian Splatting. 2024. 2, 3, 5, 7

[50] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European conference on computer vision (ECCV)*, pages 822–838, 2018. 2

[51] 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. 3, 7