I^2-SLAM: Inverting Imaging Process for Robust Photorealistic Dense SLAM

Gwangtak Bae[*1], Changwoon Choi[*1], Hyeongjun Heo[1], Sang Min Kim[1], and Young Min Kim[†1,2]

[1] Dept. of Electrical and Computer Engineering, Seoul National University
[2] INMC & IPAI, Seoul National University

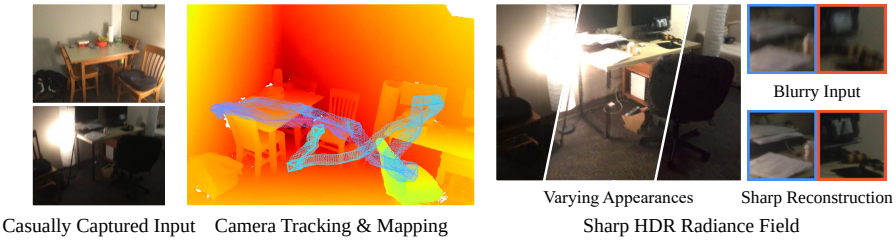


Fig. 1: We propose I^2-SLAM, a SLAM pipeline with a physical image formation process. We can reconstruct photorealistic and sharp HDR maps from casually captured videos which contain severe motion blur and varying appearances.

Abstract. We present an inverse image-formation module that can enhance the robustness of existing visual SLAM pipelines for casually captured scenarios. Casual video captures often suffer from motion blur and varying appearances, which degrade the final quality of coherent 3D visual representation. We propose integrating the physical imaging into the SLAM system, which employs linear HDR radiance maps to collect measurements. Specifically, individual frames aggregate images of multiple poses along the camera trajectory to explain prevalent motion blur in hand-held videos. Additionally, we accommodate per-frame appearance variation by dedicating explicit variables for image formation steps, namely white balance, exposure time, and camera response function. Through joint optimization of additional variables, the SLAM pipeline produces high-quality images with more accurate trajectories. Extensive experiments demonstrate that our approach can be incorporated into recent visual SLAM pipelines using various scene representations, such as neural radiance fields or Gaussian splatting. Project website

Keywords: SLAM · photorealistic 3D reconstruction · motion deblurring · HDR reconstruction

[*] Authors contributed equally to this work.
[†] Young Min Kim is the corresponding author.

1 Introduction

Simultaneous localization and mapping (SLAM) builds a map of the environment during deployment, which can be utilized in various applications, including VR/AR [10, 21], robotic navigation [6, 15, 36], and collision handling [8]. Traditional 3D SLAM approaches typically use geometric representations like points/surfels [24, 42, 50, 54, 55], mesh [3], voxel grids [4, 33], or voxel hashing [12, 35]. Recent visual SLAM approaches additionally capture visual appearances incorporating advances in Neural Radiance Fields (NeRF) [31] and its variants [25, 32]. They can synthesize photorealistic images of the environment and open up new possibilities in complex downstream tasks such as detailed semantic scene understanding [20], language-guided manipulation [45], or visual navigation [43]. Additionally, neural representations can fill unseen regions with smooth geometric estimation and require low-memory footprint [37, 47, 51]. 3D visual representations can achieve real-time performance using voxelized hash grid [32] or 3D Gaussian Splatting (3DGS) [25].

Despite many works that build visual representations using the SLAM framework, most do not maintain their performance in real-world scenarios. Casually captured videos, the standard input for visual SLAM systems, suffer from two prevalent challenges: 1) *motion blur* due to camera movement and 2) *varying appearances* resulting from auto exposure and white balancing adjustments as demonstrated in Fig. 1. The degradation in images serves as a critical bottleneck for the accuracy of the map and the pose estimation, and the error accumulates due to the incremental nature of SLAM, significantly reducing the overall quality.

This work tackles the prevailing challenges by attaching the physical image formation process that directly models the aforementioned variations. Then, we can directly optimize for the correct camera poses and raw measurements via an analysis-by-synthesis approach. Specifically, the motion-blurred image is compared against observations integrated along the estimated camera trajectories during a window of exposure time instead of an image from a single camera pose. Our pipeline approximates the camera poses as a linear movement and optimizes the start and end poses. The global trajectory estimated in the SLAM pipeline guides the initial blur movement with in-camera parameters, such as exposure time. At the same time, we match the per-frame appearance variation against simulated images and jointly optimize a differentiable tone mapper composed of exposure time, white balance function, and camera response function (CRF). We employ high dynamic range (HDR) radiance fields as a map representation to linearize the color space, which reflects the actual light intensity maps. The HDR maps simplify modeling appearance variations and produce significantly more realistic motion blur effects [13].

As our formulation, coined I^2-SLAM, inverts the actual measurement steps, the module can augment any dense visual SLAM pipelines that use image inputs, including implicit neural networks and 3D Gaussians. Our extensive experiments demonstrate that it can robustly reconstruct sharp HDR maps from RGB/RGBD streams afflicted with severe motion blur and varying appearances.

Our technical contributions can be summarized as follows:

- We present I^2-SLAM, integrating the image formation process into the visual SLAM approaches to overcome dominant challenges in real-world captures.
- Incorporating 3D maps composed of linear radiance values, our formulation jointly optimizes the approximate movement for the motion blur and tone-mapping functions for appearance variations during the SLAM framework.
- We propose an initialization and regularization method to stabilize the blur movement optimization to align with the estimated global camera trajectory, utilizing the SLAM setup.
- We enhance the robustness and performance of recent visual SLAM approaches in the real-world and our synthetic dataset, which contains severe motion blur and varying appearance.

2 Related Works

Dense Visual SLAM Visual SLAM methods use images as input and first started using multi-view geometry between sparse image features to estimate camera trajectories. DTAM [34] further demonstrated building a projective photometric cost volume, a dense map representation that unlocks the opportunity for combining various image-based applications. As deep neural networks prosper in computer vision, parts of the visual SLAM pipeline have also been successfully deployed to use latent representations [2] or assist the depth estimation [50].

Subsequent advancements in neural visual SLAM benefit from the technical innovations in neural implicit representation or improved novel representations. iMAP [47] employed implicit neural map representation for SLAM. NICE-SLAM [60] demonstrates that the multi-resolution feature grid representation can improve the speed and resolve the forgetting problem in large-scale scenes. Co-SLAM [51] explores the coordinate and parametric encodings to achieve efficient and accurate mapping. NeRF-SLAM [38] eliminates the dependency on depth by utilizing DROID-SLAM [50]. Recent results favor explicit representations when high speed and photometric quality are desired. Point-SLAM [40] creates photorealistic 3D maps using a dynamic neural point cloud. Since the introduction of 3D Gaussian Splatting [25], several concurrent works rapidly deploy them for 3D map representation for SLAM [17, 23, 29, 57].

Motion Deblurring Motion blur significantly affects the quality of many computer vision tasks and has long been investigated as an active research topic. Traditional approaches assume a convolution kernel for the motion blur and convert the operation via deconvolution [9, 14, 44, 56]. The convolution operation naturally transfers to convolutional neural networks (CNN), and more recent works estimate the non-uniform motion blur field [48], complex Fourier coefficients [5], or dense motion flow [7]. Recent neural representations [31, 32] also suffer from performance degradation when handling blurred input images. Various works account for the blur operation and reconstruct a sharp 3D map even when input images are blurred [28, 52].

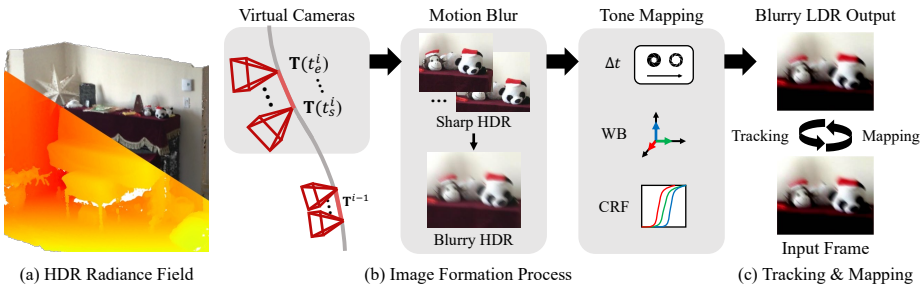


Fig. 2: Method overview. (a) We reconstruct a sharp HDR radiance field map. (b) Motion blur is simulated by integration of sharp images, which are obtained from virtual camera poses during the exposure time. Then we obtain the blurry LDR image by applying differentiable tone mapping module. (c) SLAM methods simultaneously perform tracking and mapping from degraded images to reconstruct a sharp HDR map.

The input measurements for the visual SLAM are susceptible to motion blur, as a moving camera captures an image for a duration of time. We linearize the camera trajectory during the exposure time and optimize the poses to match the blurred input, similar to [52]. However, we further exploit the SLAM setup and regularize the blur movement to be aligned with the estimated camera trajectory.

High Dynamic Range Recovery Another critical issue for high-quality visual SLAM is that the pixel values are inconsistent with different exposures in input frames. Classical works on HDR imaging demonstrate that a color space of an HDR radiance map can successfully combine multiple images of different exposures [13]. HDR radiances are linear to the scene radiance values and lead to better results in image processing and image-based modeling [7, 16, 59]. Our map representation also models the actual HDR radiance values, and we provide explicit variables to model the physical process to map the pixel values, namely the exposure time, white balance, and camera response function per frame. Recent works on novel-view synthesis and 3D reconstruction also achieve improved results as they optimize the HDR radiances with exposure times [18, 22, 30, 39]. Our work integrates factors such as exposure time into a coherent formulation for tone mapping and motion blur and stabilizes the overall optimization.

3 Method

I^2-SLAM is a generic method that can be combined with any existing photo-realistic dense SLAM methods. In Sec. 3.1, we first review the rendering techniques for two representative map representations used in our experiments: Neural Radiance Fields (NeRF) [30] and 3D Gaussian Splatting (3DGS) [25]. Then we describe our physical image formation process to render motion-blurred and appearance-varying images in Sec. 3.2. Finally, we explain how to integrate our image formation process into existing SLAM pipelines in Sec. 3.3.

3.1 Preliminaries: Rendering for Photorealistic Dense SLAM

Our approach seamlessly integrates with various methods for generating color $\mathbf{c}$ and, optionally, depth d for each pixel for a given camera pose. Owing to their photorealistic quality and ease of training, most recent visual SLAM approaches are based on variations stemming from either NeRF [31] or 3DGS [25]. This section reviews the general formulations of the two approaches, elucidating the process of generating color and depth information.

Let $\mathbf{c}(\mathbf{T}, \mathbf{p})$ and $d(\mathbf{T}, \mathbf{p})$ denote the color and depth at pixel location $\mathbf{p}$ with camera pose $\mathbf{T}$. For NeRF, we can obtain the color and depth of target pixels by marching rays and utilizing the volume rendering technique:

$$\mathbf{c}(\mathbf{T}, \mathbf{p}) = \sum_{i=1}^{N} \tau_i (1 - e^{-\sigma_i \delta_i}) \mathbf{c}_i \quad \text{where } \tau_i = e^{-\sum_{j=1}^{i-1} \sigma_j \delta_j}, \tag{1}$$

$$d(\mathbf{T}, \mathbf{p}) = \sum_{i=1}^{N} \tau_i (1 - e^{-\sigma_i \delta_i}) d_i, \tag{2}$$

where N is the number of samples for each ray, and the camera determines the ray pose $\mathbf{T}$ and pixel location $\mathbf{p}$. $\mathbf{c}_i$ and d_i are the corresponding color and depth of each sample along a ray observed from the camera center. δ is the distance between consecutive samples. Color $\mathbf{c}_i$ and volume density σ_i can be queried from MLP as in iMAP [47], hierarchical feature grid with small MLP [60], or multi-resolution hash grid [38].

3DGS [25] also adopts the volume rendering formulation, so we can train the map differentiable to input image measurements. However, 3DGS processes sparse explicit samples from 3D Gaussian primitives, which are faster. We can render images by compositing $\mathcal{N}$ ordered 3D Gaussians as follows:

$$\mathbf{c}(\mathbf{T}, \mathbf{p}) = \sum_{i=1}^{\mathcal{N}} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \text{ where } \alpha = o e^{-(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)/2}, \tag{3}$$

$$d(\mathbf{T}, \mathbf{p}) = \sum_{i=1}^{\mathcal{N}} d_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{4}$$

where $\mathbf{c}$, $o \in [0, 1]$, $\mu \in \mathbb{R}^3$, and Σ are the color, opacity, center position, and the covariance of a 3D Gaussian, respectively.

3.2 Image Formation Process

When running a visual SLAM framework, we can augment additional mapping processes from HDR radiance to image measurements. We reconstruct an HDR radiance map by making $c(\mathbf{T}, \mathbf{p})$ in Sec. 3.1 to output linear HDR color. The

HDR pixel intensity $C^i_{\text{HDR}}(\mathbf{p})$ captured during the exposure time $[t^i_s, t^i_e]$ for the ith frame is

$$C^i_{\text{HDR}}(\mathbf{p}) = \int_{t^i_s}^{t^i_e} \mathbf{c}(\mathbf{T}(t), \mathbf{p}) dt, \tag{5}$$

where $\mathbf{T}(t)$ is the camera pose at time t, $\mathbf{c}(\mathbf{T}, \mathbf{p})$ is HDR color for camera pose $\mathbf{T}$ at pixel location $\mathbf{p}$ defined in Eqs. (1) and (3). Motion blur occurs if $\mathbf{T}(t)$ changes during exposure. We numerically approximate the integral with quadrature:

$$C^i_{\text{HDR}}(\mathbf{p}) = \Delta t^i \cdot \frac{1}{N_{\text{cam}}} \sum_{j=1}^{N_{\text{cam}}} \mathbf{c}\left(\mathbf{T}\left(t^i_s + \frac{j-1}{N_{\text{cam}}-1}(t^i_e - t^i_s)\right), \mathbf{p}\right), \tag{6}$$

where N_{cam} is the number of virtual cameras to approximate the continuous integral with discrete summation and $\Delta t^i = t^i_e - t^i_s$ is exposure time. Throughout, we empirically set $N_{\text{cam}} = 5$ for experiments. If we assume that the camera velocity is constant during the short exposure time, the camera poses within time range $t \in [t^i_s, t^i_e]$ can be obtained by linearly interpolating between the start pose $\mathbf{T}(t^i_s)$ and the end pose $\mathbf{T}(t^i_e)$. We interpolate the camera pose at time t, $\mathbf{T}(t)$, by decomposing it into rotation $\mathbf{R}(t)$ and translation $\mathbf{t}(t)$:

$$\mathbf{R}(t) = \text{Slerp}\left(\mathbf{R}(t^i_s), \mathbf{R}(t^i_e), \frac{t - t^i_s}{t^i_e - t^i_s}\right), \quad \mathbf{t}(t) = \text{Lerp}\left(\mathbf{t}(t^i_s), \mathbf{t}(t^i_e), \frac{t - t^i_s}{t^i_e - t^i_s}\right), \tag{7}$$

where Slerp stands for spherical linear interpolation, Lerp is linear interpolation.

Then, the final observed color value of the pixel $\mathbf{p}$ is obtained by applying tone mapping operator Ψ^i:

$$C^i_{\text{LDR}}(\mathbf{p}) = \Psi^i(C^i_{\text{HDR}}(\mathbf{p})). \tag{8}$$

The tone mapping operator Ψ^i clips the HDR color to low dynamic range (LDR) and maps from linear color space to non-linear color space. Specifically, the tone mapping operation is composed of white balancing WB^i and camera response function CRF^i with dynamic range clipping:

$$\Psi^i(\Delta t^i \cdot \mathbf{c}) = \text{CRF}^i(\text{WB}^i(\Delta t^i \cdot \mathbf{c})). \tag{9}$$

White balance function WB^i is an element-wise product to each color channel:

$$\text{WB}^i(\mathbf{c}) = \begin{bmatrix} wb^i_r & wb^i_g & wb^i_b \end{bmatrix}^T \odot \begin{bmatrix} c_r & c_g & c_b \end{bmatrix}^T. \tag{10}$$

We parameterize non-linear CRF^i with uniformly sampled 256-dimensional grid g^i between $[0, 1]$ for each color channel. CRF should satisfy the two properties: (1) monotonically increasing function and (2) $\text{CRF}(0) = 0$ and $\text{CRF}(1) = 1$ [13]. We further adjust CRF to be physically plausible following [27]. We shift the derivatives by the smallest negative derivative and normalize CRF to satisfy the two properties. We employ differentiable grid sampling [19] to query our CRF

value. Instead of hard clipping the dynamic range between $[0, 1]$, we use a leaky clipping function with CRF to backpropagate the gradient:

$$\mathrm{CRF}_{\mathrm{leaky}}(c) = \begin{cases} \alpha c & c < 0 \\ \mathrm{CRF}(c) & 0 \le c \le 1 \\ -\frac{\alpha}{\sqrt{c}} + \alpha + 1 & 1 < c \end{cases}, \tag{11}$$

where α is a constant parameter. Throughout, we set $\alpha = 0.01$ for experiments. To summarize, our image formation process can transform the HDR radiances in our map representation into LDR camera pixel values captured with continuous exposure. The learnable parameters are the HDR radiance map, start and end poses $\mathbf{T}(t_s), \mathbf{T}(t_e)$, exposure time Δt, white balance parameters WB, and control points for CRF g. Since all modules are fully differentiable, we can optimize the parameters with a simple gradient-based optimization method by minimizing the objective function with proper regularization, as described in Sec. 3.3.

3.3 Tracking and Mapping

We optimize camera trajectories during exposure time and reconstruct a sharp HDR 3D map using the rendering loss defined by our image formation process. To optimize the camera trajectory during exposure time, we additionally propose a trajectory loss and a camera trajectory initialization method. Overall loss is sum of an image rendering loss, a depth rendering loss, and a trajectory loss:

$$\mathcal{L} = \lambda_{\mathrm{img}}\mathcal{L}_{\mathrm{img}} + \lambda_{\mathrm{depth}}\mathcal{L}_{\mathrm{depth}} + \lambda_{\mathrm{traj}}\mathcal{L}_{\mathrm{traj}}. \tag{12}$$

Image Rendering Loss We jointly optimize the in-camera parameters and the map representation within the SLAM's tracking and mapping pipeline by applying an image rendering loss function as follows:

$$\mathcal{L}_{\mathrm{img}} = \sum_{i=1}^{N} \sum_{\mathbf{p}} |C^i_{\mathrm{LDR}}(\mathbf{p}) - \hat{C}^i(\mathbf{p})|, \tag{13}$$

where $C^i_{\mathrm{LDR}}(\mathbf{p})$ is a rendered color output from our image formation process and $\hat{C}^i(\mathbf{p})$ is the observed color at pixel location $\mathbf{p}$ of frame i. Even if the input images are degraded, the proposed rendering loss accounts for the physical degradation process and can reconstruct a sharp HDR radiance map.

Depth Rendering Loss We constrain the depth camera pose to be aligned with the color camera's trajectory during exposure time. Specifically, we assign a pose with a minimum depth error as the depth camera pose and apply depth rendering loss for the selected pose as follows:

$$\mathcal{L}_{\mathrm{depth}} = \sum_{i=1}^{N} \sum_{\mathbf{p}} |d(\mathbf{T}(t^i_{\mathrm{d}}), \mathbf{p}) - \hat{d}^i(\mathbf{p})|, \text{where } t^i_{\mathrm{d}} = \underset{t \in [t^i_s, t^i_e]}{\arg\min} |d(\mathbf{T}(t), \mathbf{p}) - \hat{d}^i(\mathbf{p})|, \tag{14}$$

where $d(\mathbf{T}(t_{\mathrm{d}}^i), \mathbf{p})$ is a rendered depth from the depth camera pose $\mathbf{T}(t_{\mathrm{d}}^i)$ and $\hat{d}^i(\mathbf{p})$ is the ground-truth depth at pixel location $\mathbf{p}$ of frame i. Note that the depth rendering loss affects the color camera's trajectory since the depth camera pose is interpolated from $\mathbf{T}(t_s^i)$ and $\mathbf{T}(t_e^i)$. Depth rendering loss ensures that at least one pose along the camera poses during the exposure time outputs accurate depth rendering.

Unlike the integration process for color pixels, we optimize a single pose for a depth camera, assuming that the depth information is captured at a single moment within the time window. Although the depth camera measures some inaccurate depth values in fast-moving scenarios, most of the noises are filtered as invalid pixels by sensor manufacturers [41]. Depth sensors usually output invalid pixels on object boundaries and black objects, and most RGBD-SLAM methods ignore those regions.

Trajectory Regularization We propose a trajectory loss that regularizes the camera trajectory during the exposure time. We design the trajectory loss function with two insights. First, the camera poses during the exposure time should be aligned with the global trajectory. As SLAM progressively optimizes the camera poses, the global trajectory can be obtained from previous localization results without additional cost and it gives meaningful information to the temporal sensor movement. We estimate the global trajectory by connecting the pose of the midpoint during each exposure time in the previous frames. Second, the size of the motion blur kernel is determined by the temporal velocity and the exposure time. Namely, the longer the exposure time Δt^i and the faster the temporal velocity, the further the distance between the start and end camera poses.

Our trajectory loss regularizes the temporal camera trajectory during the exposure time to be aligned with the global trajectory and the length of it to be proportional to the exposure time and temporal velocity, assuming piecewise linear velocity:

$$\mathcal{L}_{\mathrm{traj}} = \mathcal{L}_{\mathrm{traj}}^{\mathbf{t}} + \mathcal{L}_{\mathrm{traj}}^{\mathbf{R}}, \tag{15}$$

$$\mathcal{L}_{\mathrm{traj}}^{\mathbf{t}} = \left\| \mathbf{t}(t_e^{i-1}) - \mathrm{Lerp}(\mathbf{t}^{i-1}, \mathbf{t}^i, a\Delta t^{i-1}) \right\|_2^2 + \left\| \mathbf{t}(t_s^i) - \mathrm{Lerp}(\mathbf{t}^{i-1}, \mathbf{t}^i, 1 - a\Delta t^i) \right\|_2^2, \tag{16}$$

$$\mathcal{L}_{\mathrm{traj}}^{\mathbf{R}} = \left\| \mathbf{R}(t_e^{i-1}) - \mathrm{Slerp}(\mathbf{R}^{i-1}, \mathbf{R}^i, a\Delta t^{i-1}) \right\|_2^2 + \left\| \mathbf{R}(t_s^i) - \mathrm{Slerp}(\mathbf{R}^{i-1}, \mathbf{R}^i, 1 - a\Delta t^i) \right\|_2^2, \tag{17}$$

where $\mathbf{t}$ and $\mathbf{R}$ are the translation and rotation vector of camera pose $\mathbf{T}$, and a is an unknown global scale parameter that is related to the input frame rate. $\mathbf{t}^i$ and $\mathbf{R}^i$ are the center pose between the start and end pose of frame i. The scale parameter a is also jointly optimized within the tracking process. We further describe the relation between our scale parameter a and the input frame rate in the supplementary material.

We also propose an initialization strategy for robust optimization exploiting the global trajectory. With a constant velocity assumption, we initialize the $i+1$th frame's camera poses $\mathbf{T}(t_s^{i+1})$ and $\mathbf{T}(t_e^{i+1})$ by extrapolating estimated camera poses of i and $i-1$th frames. We initialize the start and end poses to be separated with a small predefined distance along the global trajectory.

4 Experiments

We demonstrate that I^2-SLAM can reconstruct a sharp HDR radiance map from casually captured videos with various input modalities. We describe our experimental setup in challenging datasets in Sec. 4.1. We show that I^2-SLAM enhances state-of-the-art dense visual SLAM methods in Sec. 4.2. We conduct ablation studies and runtime analyses in Sec. 4.3 and Sec. 4.4, respectively.

4.1 Experimental Setup

Baselines I^2-SLAM serves as a versatile module that can be attached to dense visual SLAM methods to improve its performance. We apply our approach to the state-of-the-art RGB-SLAM method, NeRF-SLAM [38], which employs NeRF as a map representation and uses DROID-SLAM [50] as a tracking backbone. We re-implemented NeRF-SLAM with torch-ngp [49] and notate as NeRF-SLAM†. NeRF-SLAM† uses same loss functions of NeRF-SLAM. We additionally test I^2-SLAM with an RGBD-SLAM approach to tackle challenging sequences in ScanNet [11] where robust learning-based SLAM method, DROID-SLAM, often fails without using additional depth channel. We employ a 3DGS-based RGBD-SLAM method, SplaTAM [23], to test I^2-SLAM in RGBD inputs.

Datasets Most of the existing synthetic datasets for SLAM evaluation assume the ideal capturing setup, and do not exhibit any camera motion blur or dynamic appearance changes. We therefore propose a new dataset incorporating these effects. The new dataset contains realistic image degradation by simulating motion blur and auto exposure. We render the images and depth information with Cycles path tracer [1]. Also, we test I^2-SLAM on challenging real-world datasets. We use TUM-RGBD [46] and ScanNet [11] dataset for evaluating RGB and RGBD scenarios, respectively.

Evaluation We report the average values of three runs with different random seeds for all the quantitative evaluations except the ablation study and the runtime analysis. We measure PSNR, SSIM [53], and LPIPS [58] between rendered images from the reconstructed map and sharp ground-truth images which can be obtained from our synthetic dataset. Since there are no ground-truth sharp images in the real-world dataset, we evaluate the view synthesis performance only for the sharp frames that are manually annotated. Also, we run test-time optimization [26] to factor out pose errors in measuring the rendering quality of RGB-SLAMs. We report ATE RMSE [46] for tracking performance evaluation.

Table 1: Rendering quality comparison against the RGB-SLAM baseline on TUM-RGBD [46] and synthetic dataset. I^2-SLAM represents our RGB-SLAM model which is incorporated into our re-implementation of NeRF-SLAM [38], NeRF-SLAM†.

Methods	Metrics	TUM-RGBD [46]			Synthetic				
		fr1/desk	fr2/xyz	fr3/office	SP	LOU	IF0	IF1	IF2
NeRF-SLAM† [38]	PSNR	25.97	29.97	24.72	28.64	25.43	30.20	26.09	26.70
	SSIM	0.825	0.900	0.727	0.810	0.832	0.867	0.789	0.842
	LPIPS	0.222	0.093	0.366	0.328	0.323	0.327	0.302	0.270
	Depth L1	10.97	**17.83**	30.92	40.68	55.80	20.72	5.02	19.76
I^2-SLAM	PSNR	**27.23**	**32.06**	**28.91**	**28.99**	**27.59**	**32.33**	**30.16**	**28.89**
	SSIM	**0.835**	**0.916**	**0.833**	**0.827**	**0.875**	**0.902**	**0.898**	**0.887**
	LPIPS	**0.186**	**0.074**	**0.193**	**0.284**	**0.260**	**0.286**	**0.211**	**0.241**
	Depth L1	**9.04**	17.94	**17.60**	**20.33**	**41.92**	**20.48**	**3.55**	**17.28**

Table 2: Rendering quality comparison against the RGBD-SLAM baseline on ScanNet [11] and synthetic dataset. I^2-SLAM in this table represents our RGBD-SLAM model which is incorporated into SplaTAM [23].

Methods	Metrics	ScanNet [11]				Synthetic				
		0024-01	0031-00	0736-00	0785-00	SP	LOU	IF0	IF1	IF2
SplaTAM [23]	PSNR	21.60	24.64	**24.50**	19.63	21.38	19.78	**24.22**	22.36	23.82
	SSIM	**0.786**	0.773	**0.847**	0.719	0.820	**0.790**	**0.855**	0.766	0.824
	LPIPS	0.236	0.275	0.182	0.340	0.232	0.246	**0.227**	0.281	0.230
I^2-SLAM	PSNR	**23.39**	**26.89**	24.07	**26.40**	**26.18**	**21.98**	23.88	**23.72**	**24.07**
	SSIM	0.780	**0.796**	0.828	**0.762**	**0.842**	0.770	0.798	**0.796**	**0.826**
	LPIPS	**0.180**	**0.236**	**0.175**	**0.238**	**0.193**	**0.231**	0.263	**0.233**	**0.205**

We use the center of camera poses $\mathbf{T}(t_s^i)$ and $\mathbf{T}(t_e^i)$ to evaluate our method and use scale-aligned ground-truth trajectory for RGB-SLAM to handle the scale ambiguity. Further details can be found in the supplementary material.

4.2 Experimental Results

Quantitative Results We report the quantitative results of map rendering performance for keyframes of RGB and RGBD datasets in Tabs. 1 and 2, respectively. We observe that I^2-SLAM enhances the rendering quality of NeRF-SLAM† across all metrics on the both synthetic and real-world TUM-RGBD [46] datasets. I^2-SLAM also substantially enhances the depth accuracy for RGB-SLAM. In the RGBD dataset scenario, I^2-SLAM also shows superior rendering quality in most scenes when attached to SplaTAM [23]. Especially, I^2-SLAM improves the view-synthesis performance of SplaTAM in a large margin in SP of synthetic dataset and 0785-00 of ScanNet [11] dataset, whose appearances are changing dynamically within the videos. It shows that our in-camera model is especially advantageous when combined with the models without view-dependent effects. Furthermore, we report the camera trajectory error of RGB-SLAM methods in Tab. 3. The results show that I^2-SLAM substantially improves the tracking performance of NeRF-SLAM† even if NeRF-SLAM† takes accurate initial tra-

Table 3: We evaluate the tracking performance of RGB-SLAM methods on TUM-RGBD [46] and synthetic dataset in terms of ATE-RMSE (cm).

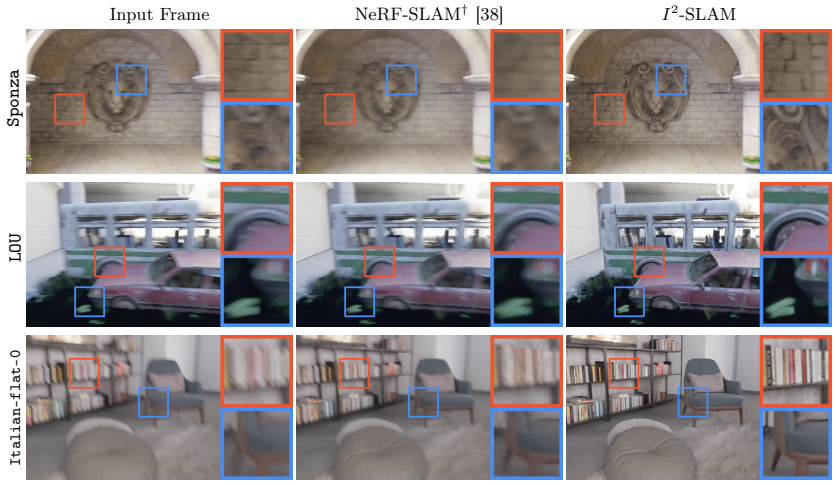
Methods	TUM-RGBD [46]			Synthetic				
	fr1/desk	fr2/xyz	fr3/office	SP	LOU	IF0	IF1	IF2
NeRF-SLAM† [38]	2.08	0.41	7.13	3.97	**3.20**	3.38	1.14	0.65
I^2-SLAM	**1.64**	**0.26**	**1.95**	**1.50**	3.23	**1.59**	**0.74**	**0.33**

Table 4: Tracking accuracy comparison against the RGBD-SLAM baseline on ScanNet [11] and synthetic dataset. ATE-RMSE (cm) is measured as an evaluation metric.

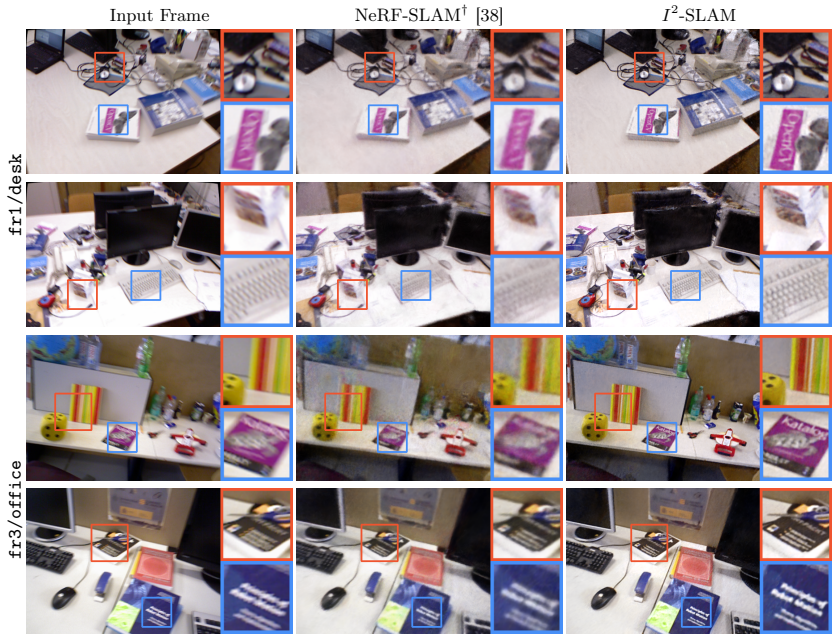
Methods	ScanNet [11]				Synthetic				
	0024-01	0031-00	0736-00	0785-00	SP	LOU	IF0	IF1	IF2
SplaTAM [23]	1.80	**3.01**	1.13	5.91	1.11	**1.50**	2.04	**1.02**	**0.61**
I^2-SLAM	**1.41**	3.25	**1.00**	**4.59**	**0.86**	1.55	**1.96**	1.05	0.87

jectory from DROID-SLAM as initial camera poses. It supports our claim that image formation process, when appropriately modeled, can enhance the tracking accuracy of inputs that are casually captured. In RGBD datasets, however, the trajectory accuracy improvement is subtle since the depth information mostly determines the camera poses as reported in Tab. 4.

Qualitative Results We demonstrate the rendered images of I^2-SLAM with NeRF-SLAM† in Fig. 3. As shown in Fig. 3 (a), I^2-SLAM successfully renders sharp images, whereas NeRF-SLAM† has blurry artifacts come from the degraded input, for example, bricks in `Sponza`, boundary of the tire in `LOU`, complex texture of books and thin legs of the chair in `Italian-flat-0`. More importantly, our approach also shows remarkable performance in reconstructing sharp maps from TUM-RGBD data which contains the real-world camera motion. The texts on the books and objects on the table are clearly rendered with I^2-SLAM when we compare to the results of baseline and blurred input frames. In Fig. 4, we show samples of rendered images in RGBD scenario. We can observe the consistent improvement over SplaTAM in generating a sharp map in our synthetic dataset. The letters on the book cover and the bricks are clearly rendered with I^2-SLAM. Also, we observe significant map quality enhancement in the ScanNet dataset. The letters on the board and tassels in `0024-01`, the detection mark and objects on the table in `0736-00`, and the pictures in `0785-00` are deblurred with our approach. Furthermore, since the color of Gaussians in SplaTAM is consistent regardless of the exposure of images, SplaTAM does not appropriately model the brightness changes as we can see in the darker color of bricks in `Sponza` and tiled artifacts in `0785-00`. Such artifacts are removed by modeling the tone mapping process.



(a) Synthetic Dataset



(b) TUM-RGBD [46]

Fig. 3: Qualitative results on applying I^2-SLAM to the RGB-SLAM method.

4.3 Ablation Study

We conduct an ablation study in ScanNet [11] dataset. Table 5 shows that each component plays a crucial role in improving tracking or mapping performance. In Tab. 5, Traj. Reg. refers to the trajectory regularization in Sec. 3.3. Motion

(a) Synthetic Dataset

(b) ScanNet [11]

Fig. 4: Qualitative results on applying I^2-SLAM to the RGBD-SLAM method.

blur-aware rendering improves tracking performance by preventing the accumulation of tracking errors. HDR map shows improved rendering quality over LDR representations. Trajectory regularization affects multi-camera optimization, resulting in enhancements in both tracking and mapping.

4.4 Runtime analysis

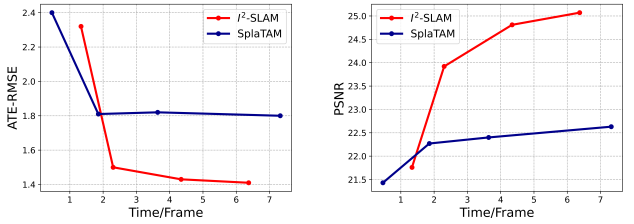
We analyze how much our approach affects the runtime and performance with ScanNet `0024-01` in Tab. 6. I^2-SLAM exhibits better ATE-RMSE and PSNR

Table 5: Ablation study.

Traj. Reg.	Tone Mapping	Motion Blur	ATE	PSNR	SSIM	LPIPS
✓	✓	✓	**2.56**	**25.62**	**0.801**	**0.195**
✗	✓	✓	2.66	24.80	0.769	0.203
✓	✗	✓	2.60	22.39	0.756	0.226
✗	✗	✓	2.63	22.36	0.755	0.228
✗	✗	✗	2.71	23.05	0.793	0.235

Table 6: Runtime analysis.

	Mapping /Frame	Tracking /Frame	ATE	PSNR	SSIM	LPIPS
I^2-SLAM	2.058s	8.494s	1.42	24.16	0.786	0.211
I^2-SLAM-S	0.436s	1.867s	1.50	23.92	0.795	0.230
SplaTAM [23]	0.424s	1.433s	1.81	22.26	0.800	0.224
SplaTAM-S [23]	0.086s	0.377s	2.40	21.43	0.777	0.243



(a) ATE-RMSE over iteration time **(b)** PSNR over iteration time

Fig. 5: Performance variations over iteration time of I^2-SLAM and SplaTAM [23].

metrics compared to SplaTAM [23], but it comes at the cost of slower speed. However, I^2-SLAM-S, which uses 20% iterations, takes a similar runtime but yields better performance. In contrast, SplaTAM-S, which also uses 20% iterations, shows a significant decrease in performance.

In Fig. 5, we analyze the trend of performance changes concerning the number of iterations. Four models with different numbers of iterations are compared. In most cases, our method demonstrates better tracking and rendering performance when using a similar runtime.

5 Conclusion

We present I^2-SLAM, a generic module that improves the quality of existing visual SLAM approaches by inverting the image formation process for casually captured videos. We show the effectiveness of our module by incorporating it into the state-of-the-art methods in RGB and RGBD-SLAM approaches. With I^2-SLAM, existing visual SLAM pipelines can effectively handle the varying appearance and motion blur that are prevalent in-the-wild capturing scenarios.

Although I^2-SLAM is able to reconstruct sharp HDR maps while estimating the camera trajectory from casually captured videos, there are some limitations. The motion blur simulation with the sum of multiple cameras accompanies the longer rendering and optimization time. An efficient approximation method to simulate camera motion blur would be an interesting research direction to improve our approach.

Acknowledgements This work is supported by the National Research Foundation of Korea(NRF) grant (No. RS-2023-00218601) and IITP grant [NO.RS-

2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)] funded by the Korea government(MSIT).

References

1. Blender, O.C.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018), http://www.blender.org
2. Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., Davison, A.J.: Codeslam—learning a compact, optimisable representation for dense visual slam. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2560–2568 (2018)
3. Bloesch, M., Laidlow, T., Clark, R., Leutenegger, S., Davison, A.J.: Learning meshes for dense visual slam. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2019)
4. Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D.: Real-time camera tracking and 3d reconstruction using signed distance functions. In: Robotics: Science and Systems. vol. 2, p. 2 (2013)
5. Chakrabarti, A.: A neural approach to blind motion deblurring. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. pp. 221–235. Springer (2016)
6. Chaplot, D.S., Gandhi, D., Gupta, S., Gupta, A., Salakhutdinov, R.: Learning to explore using active neural slam. In: International Conference on Learning Representations (ICLR) (2020)
7. Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3291–3300 (2018)
8. Chen, T., Culbertson, P., Schwager, M.: Catnips: Collision avoidance through neural implicit probabilistic scenes. arXiv preprint arXiv:2302.12931 (2023)
9. Cho, S., Lee, S.: Fast motion deblurring. In: ACM SIGGRAPH Asia 2009 papers, pp. 1–8 (2009)
10. Covolan, J.P.M., Sementille, A.C., Sanches, S.R.R.: A mapping of visual slam algorithms and their applications in augmented reality. In: 2020 22nd Symposium on Virtual and Augmented Reality (SVR). pp. 20–29. IEEE (2020)
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2017)
12. Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. ACM Transactions on Graphics (ToG) **36**(4), 1 (2017)
13. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. p. 369–378. SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., USA (1997). https://doi.org/10.1145/258734.258884
14. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. In: Acm Siggraph 2006 Papers, pp. 787–794 (2006)
15. Gupta, S., Davidson, J., Levine, S., Sukthankar, R., Malik, J.: Cognitive mapping and planning for visual navigation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2616–2625 (2017)

16. Hu, Y., He, H., Xu, C., Wang, B., Lin, S.: Exposure: A white-box photo post-processing framework. ACM Transactions on Graphics (TOG) **37**(2), 1–17 (2018)
17. Huang, H., Li, L., Cheng, H., Yeung, S.K.: Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras. arXiv preprint arXiv:2311.16728 (2023)
18. Huang, X., Zhang, Q., Feng, Y., Li, H., Wang, X., Wang, Q.: Hdr-nerf: High dynamic range neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18398–18408 (2022)
19. Jaderberg, M., Simonyan, K., Zisserman, A., kavukcuoglu, k.: Spatial transformer networks. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015), https://proceedings.neurips.cc/paper_files/paper/2015/file/33ceb07bf4eeb3da587e268d663aba1a-Paper.pdf
20. Jatavallabhula, K.M., Kuwajerwala, A., Gu, Q., Omama, M., Chen, T., Maalouf, A., Li, S., Iyer, G., Saryazdi, S., Keetha, N., et al.: Conceptfusion: Open-set multimodal 3d mapping. arXiv preprint arXiv:2302.07241 (2023)
21. Jinyu, L., Bangbang, Y., Danpeng, C., Nan, W., Guofeng, Z., Hujun, B.: Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality. Virtual Reality & Intelligent Hardware **1**(4), 386–410 (2019)
22. Jun-Seong, K., Yu-Ji, K., Ye-Bin, M., Oh, T.H.: Hdr-plenoxels: Self-calibrating high dynamic range radiance fields. In: European Conference on Computer Vision. pp. 384–401. Springer (2022)
23. Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. arXiv preprint arXiv:2312.02126 (2023)
24. Keller, M., Lefloch, D., Lambers, M., Izadi, S., Weyrich, T., Kolb, A.: Real-time 3d reconstruction in dynamic scenes using point-based fusion. In: 2013 International Conference on 3D Vision-3DV 2013. pp. 1–8. IEEE (2013)
25. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)
26. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5741–5751 (2021)
27. Liu, Y.L., Lai, W.S., Chen, Y.S., Kao, Y.L., Yang, M.H., Chuang, Y.Y., Huang, J.B.: Single-image hdr reconstruction by learning to reverse the camera pipeline. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1651–1660 (2020)
28. Ma, L., Li, X., Liao, J., Zhang, Q., Wang, X., Wang, J., Sander, P.V.: Deblur-nerf: Neural radiance fields from blurry images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12861–12870 (2022)
29. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. arXiv preprint arXiv:2312.06741 (2023)
30. Mildenhall, B., Hedman, P., Martin-Brualla, R., Srinivasan, P.P., Barron, J.T.: Nerf in the dark: High dynamic range view synthesis from noisy raw images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16190–16199 (2022)
31. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)

32. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
33. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE international symposium on mixed and augmented reality. pp. 127–136. Ieee (2011)
34. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: Dense tracking and mapping in real-time. In: 2011 international conference on computer vision. pp. 2320–2327. IEEE (2011)
35. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. ACM Transactions on Graphics (ToG) **32**(6), 1–11 (2013)
36. Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., Nieto, J.: Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1366–1373. IEEE (2017)
37. Ortiz, J., Clegg, A., Dong, J., Sucar, E., Novotny, D., Zollhoefer, M., Mukadam, M.: isdf: Real-time neural signed distance fields for robot perception. In: Robotics: Science and Systems (2022)
38. Rosinol, A., Leonard, J.J., Carlone, L.: Nerf-slam: Real-time dense monocular slam with neural radiance fields. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3437–3444. IEEE (2023)
39. Rückert, D., Franke, L., Stamminger, M.: Adop: Approximate differentiable one-pixel point rendering. ACM Transactions on Graphics (ToG) **41**(4), 1–14 (2022)
40. Sandström, E., Li, Y., Van Gool, L., Oswald, M.R.: Point-slam: Dense neural point cloud-based slam. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18433–18444 (2023)
41. Sarbolandi, H., Lefloch, D., Kolb, A.: Kinect range sensing: Structured-light versus time-of-flight kinect. Computer vision and image understanding **139**, 1–20 (2015)
42. Schops, T., Sattler, T., Pollefeys, M.: Bad slam: Bundle adjusted direct rgb-d slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 134–144 (2019)
43. Shafiullah, N.M.M., Paxton, C., Pinto, L., Chintala, S., Szlam, A.: Clip-fields: Weakly supervised semantic fields for robotic memory. arXiv preprint arXiv:2210.05663 (2022)
44. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. Acm transactions on graphics (tog) **27**(3), 1–10 (2008)
45. Shen, W., Yang, G., Yu, A., Wong, J., Kaelbling, L.P., Isola, P.: Distilled feature fields enable few-shot language-guided manipulation. In: 7th Annual Conference on Robot Learning (2023)
46. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. pp. 573–580. IEEE (2012)
47. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: imap: Implicit mapping and positioning in real-time. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6229–6238 (2021)
48. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 769–777 (2015)

49. Tang, J.: Torch-ngp: a pytorch implementation of instant-ngp (2022), https://github.com/ashawkey/torch-ngp
50. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. Advances in neural information processing systems **34**, 16558–16569 (2021)
51. Wang, H., Wang, J., Agapito, L.: 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. pp. 13293–13302 (2023)
52. Wang, P., Zhao, L., Ma, R., Liu, P.: Bad-nerf: Bundle adjusted deblur neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4170–4179 (2023)
53. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
54. Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., Davison, A.: Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems (2015)
55. Whelan, T., Salas-Moreno, R.F., Glocker, B., Davison, A.J., Leutenegger, S.: Elasticfusion: Real-time dense slam and light source estimation. The International Journal of Robotics Research **35**(14), 1697–1716 (2016)
56. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. International journal of computer vision **98**, 168–186 (2012)
57. Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: Gaussian-slam: Photo-realistic dense slam with gaussian splatting. arXiv preprint arXiv:2312.10070 (2023)
58. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
59. Zhang, X., Matzen, K., Nguyen, V., Yao, D., Zhang, Y., Ng, R.: Synthetic defocus and look-ahead autofocus for casual videography. arXiv preprint arXiv:1905.06326 (2019)
60. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: Nice-slam: Neural implicit scalable encoding for slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12786–12796 (2022)