CG-SLAM: Efficient Dense RGB-D SLAM in a Consistent Uncertainty-aware 3D Gaussian Field

Jiarui Hu[1*], Xianhao Chen[2*], Boyin Feng[1], Guanglin Li[1], Liangjing Yang[2], Hujun Bao[1], Guofeng Zhang[1], and Zhaopeng Cui[1†]

[1] State Key Lab of CAD&CG, Zhejiang University
[2] ZJU-UIUC Institute, International Campus, Zhejiang University

Abstract. Recently neural radiance fields (NeRF) have been widely exploited as 3D representations for dense simultaneous localization and mapping (SLAM). Despite their notable successes in surface modeling and novel view synthesis, existing NeRF-based methods are hindered by their computationally intensive and time-consuming volume rendering pipeline. This paper presents an efficient dense RGB-D SLAM system, *i.e.*, CG-SLAM, based on a novel uncertainty-aware 3D Gaussian field with high consistency and geometric stability. Through an in-depth analysis of Gaussian Splatting, we propose several techniques to construct a consistent and stable 3D Gaussian field suitable for tracking and mapping. Additionally, a novel depth uncertainty model is proposed to ensure the selection of valuable Gaussian primitives during optimization, thereby improving tracking efficiency and accuracy. Experiments on various datasets demonstrate that CG-SLAM achieves superior tracking and mapping performance with a notable tracking speed of around 15 Hz. We will make our source code publicly available. Project page: https://zju3dv.github.io/cg-slam.

Keywords: Dense Visual SLAM · Neural Rendering · 3D Gaussian field

1 Introduction

Dense visual Localization and Mapping (Visual SLAM) is a long-standing problem in 3D computer vision over recent decades, which targets performing pose tracking and scene mapping simultaneously with a variety of downstream applications such as virtual/augmented reality (VR/AR), robot navigation, and autonomous driving. Traditional visual SLAM systems [24] have shown accurate tracking performance across various scenes, while the underlying 3D representations (*e.g.*, point cloud, mesh, and surfel) demonstrate limitations in facilitating highly free scene exploration, such as photorealistic scene touring, fine-grained map updating, *etc.*

Inspired by the Neural Radiance Field (NeRF) [29] in surface reconstruction and view rendering, some novel NeRF-based SLAM methods [17, 28, 37, 61] have

*Jiarui Hu and Xianhao Chen contributed equally to this work.
†Corresponding authors.

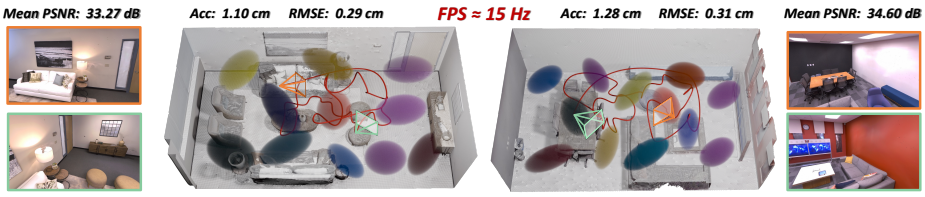

Fig. 1: CG-SLAM, which adopts a well-designed 3D Gaussian field, can simultaneously achieve state-of-the-art performance in localization, reconstruction and rendering. Benefiting from 3D Gaussian representation and a new GPU-accelerated framework that is developed from a thorough derivative analysis of camera pose in 3D Gaussian Splatting [23], CG-SLAM can perform extremely fast rendering and solve the long-standing efficiency bottleneck suffered by previous rendering-based SLAM methods.

been proposed recently and demonstrated promising performance in tracking, surface modeling, and novel view synthesis. Nevertheless, existing NeRF-based methods follow the ray-tracing rendering pipeline that is computation-intensive and time-consuming. Therefore they can only perform tracking and mapping by sampling a limited number of camera rays, ignoring the natural structural information in images. To avoid local optima in tracking and artifacts, they normally require many optimization steps, which makes it struggle to bring the best of both worlds concerning accuracy and efficiency.

Very recently the 3D Gaussian Splatting [23] method has been introduced for efficient novel view synthesis, and its rasterization-based rendering pipeline allows much faster image-level rendering, showing great potential in solving the inherent challenge of NeRF-based SLAM. However, it is nontrivial to reasonably incorporate the 3D Gaussian field in the SLAM setting. As a photorealistic view synthesis technique, the 3D Gaussian field is prone to overfitting the input images due to strong anisotropy and the lack of explicit multi-view constraints. As a result, on one hand, the 3D Gaussian splatting can not guarantee accurate modeling of 3D surfaces; on the other hand, since the Gaussians may not align with the environment's surfaces, this will lead to poor extrapolation capability and further degrade the camera tracking. Moreover, the increase of Gaussians in the mapping process will inevitably slow down the tracking efficiency.

In this paper, we introduce a real-time Gaussian splatting SLAM system, *i.e.*, CG-SLAM, based on a novel uncertainty-aware 3D Gaussian field with high consistency and geometric stability. To this end, we first conduct a comprehensive mathematical analysis regarding the derivatives of camera poses in the EWA (Elliptical Weighted Average) splatting process [63], and develop a CUDA framework tailored for the SLAM task that effectively decouples the tracking and mapping components. Second, in order to reduce the inherent overfitting problem, we utilize a scale regularization term that appropriately encourages the Gaussian ellipsoids to approximate Gaussian spheres, to reduce anisotropy and achieve a good trade-off between tracking accuracy and rendering realism. At the same time, we observed that solely employing alpha-blending depth cannot

impose effective constraints on the positions of Gaussian primitives. Thus, towards high-quality mapping, we further align median depth and alpha-blending depth to encourage Gaussian primitives to be well distributed over the scene surfaces, facilitating a consistent Gaussian field with more concentrated geometry density. Furthermore, in order to further improve the system's accuracy and efficiency, we design a novel depth uncertainty model to guide our Gaussian-based SLAM to focus on those stable and informative ones. We evaluate our system on a wide variety of RGB-D datasets, and the experimental results demonstrate that our CG-SLAM has superior performance in terms of tracking accuracy, reconstruction quality, and runtime efficiency.

Overall, our contributions can be summarized as follows:

- We present a new GPU-accelerated framework for real-time dense RGB-D SLAM based on a thorough theoretical analysis of camera pose derivatives in 3D Gaussian Splatting.
- We design multiple loss terms to build up a consistent and stable 3D Gaussian field suitable for tracking and mapping.
- We propose a novel depth uncertainty model, which assists our system in selecting more valuable Gaussian primitives during optimization, thereby improving tracking efficiency and accuracy.
- Experiments on various datasets demonstrate that our method can achieve competitive or better tracking and mapping results compared to baselines.

2 Related Work

2.1 Dense Visual SLAM

Following the seminal contributions of DTAM [32] and KinectFusion [31] to dense visual SLAM systems, there has been significant progress in developing efficient scene representation models. TSDF [31, 32], surfels [22, 39, 52], voxel hashing [6, 33, 34], and octrees [42, 49] have been introduced to tackle the challenges of scalability within the SLAM task. Some more advanced technologies, including bundle adjustment [39], loop closure, and learning-based algorithms [25,26,46,47,55], were subsequently integrated into the SLAM framework to further improve system performance. These enhancements have significantly brought better accuracy and robustness for localization and reconstruction capabilities, thus pushing the frontiers of what is achievable in dense SLAM systems. Compared with traditional methods, our CG-SLAM can reconstruct a fully dense 3D map for rich applications.

2.2 Neural Implict Radiance Field based SLAM

Neural radiance fields [29] have shown promising potential in many 3D computer vision applications, such as novel view synthesis [3, 4, 14, 30], dynamic scene modeling [13, 15, 41, 48], and generalization [18, 40, 51, 53, 58]. Recent research works [27, 45] attempted to replace traditional maps, in tasks such as structure

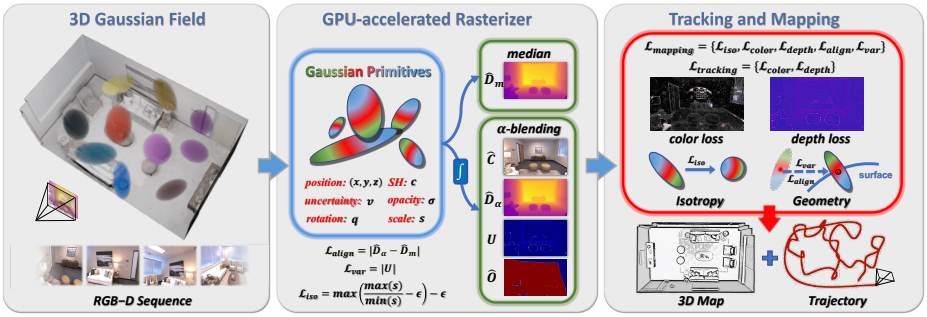

Fig. 2: System Overview. In a 3D Gaussian field constructed from an RGB-D sequence, we can render color, depth, opacity, and uncertainty maps through a GPU-accelerated rasterizer. Additionally, we attach a new uncertainty property to each Gaussian primitive to filter informative primitives. In the mapping process, we utilize multiple rendering results to design effective loss functions towards a consistent and stable Gaussian field. Subsequently, we employ appearance and geometry cues to perform accurate and efficient tracking.

from motion (SFM) and SLAM, with the neural implicit field to jointly optimize scene representation and camera poses. Different kinds of neural fields brought insights into NeRF-SLAM works. NICE-SLAM [61] chose a fully covered voxel grid to store neural features, while Vox-Fusion [56] further improved this grid to an adaptive size. Besides, Point-SLAM [37] attached feature embeddings to the point cloud on object surfaces. The neural point-based method is more flexible and can encode more concentrated volume density. Co-SLAM [50] adopted a hybrid representation that includes coordinate encoding and hash grids to achieve smoother reconstruction and faster convergence. In addition to the aforementioned works, some methods [8, 20, 36] only used the neural field as a map and still performed tracking based on a traditional feature point-based visual odometry. Our CG-SLAM system can achieve better and more efficient performance in tracking, mapping, and rendering than NeRF-based methods.

2.3 3D Gaussian Splatting Field

3D Gaussian splatting is a revolutionary novel-view synthesis approach in 3D computer vision. This approach does not contain any neural network and allows photorealistic real-time($\geq$100 FPS) rendering at 1080p resolution. 3D Gaussian splatting has influenced a wide range of research topics, such as the 3D avatar [1, 19, 35, 59, 62], scene editing [7, 12, 57], image segment [60], and surface reconstruction [5, 11, 16]. These studies have demonstrated its generalization and effectiveness. Similar to the neural field, some concurrent manuscripts [21, 54] have been made to reversely infer camera poses within a built 3D Gaussian field. However, these works straightforwardly apply a raw 3D Gaussian field in the SLAM framework without specialized designs, such as anisotropy regularization and uninformative Gaussian primitive filtering. Additionally, they overlook

the design on efficiency, which is the most important improvement that the Gaussian splatting technique should bring to a SLAM system. We believe that the advantages of the 3D Gaussian field in pose optimization have not been fully explored in these works, especially in terms of efficiency, and expect to further develop an advanced Gaussian-based visual SLAM system.

3 Method

The overview of our proposed rasterization-based Gaussian SLAM system is shown in Fig. 2. Given a set of RGB-D sequences, our system incrementally generates a stable, consistent, and uncertainty-aware Gaussian field, serving camera pose optimization and geometry reconstruction. In Sec. 3.1, we briefly introduce the 3D Gaussian splatting model and rasterization principles. We incorporated an uncertainty model that utilizes the geometry prior to attach the uncertainty property on rendered images and Gaussian primitives. This strategy helps remove outliers in mapping and makes full use of informative Gaussians in tracking (Sec. 3.2). Moreover, in Sec. 3.3, we detail the Gaussian primitive management strategy and some innovative loss terms that ensure geometry stability and accuracy. Finally, by minimizing the re-rendering loss from low-uncertainty primitives, we can build a real-time and accurate tracking module (Sec. 3.4).

3.1 Preliminary

Scene Representation. 3D Gaussian Splatting defines a 3D scene as a set of anisotropic Gaussian distributions, which are associated with means $\mathbf{X} \in \mathbb{R}^3$ and covariances $\mathbf{\Sigma} \in \mathbb{R}^{3\times3}$. To ensure that the covariance matrix remains positive semi-definite throughout the gradient descent, it is endowed with a more intuitive and comprehensible physical meaning, that is, the configuration of an ellipsoid. Specifically, $\mathbf{\Sigma}$ is simplified and decomposed into:

$$\mathbf{\Sigma} = \mathbf{R}\mathbf{S}\mathbf{S}^T\mathbf{R}^T \ , \tag{1}$$

where scaling matrix $\mathbf{S} = diag([\mathbf{s}])$ is derived from the scale factor $\mathbf{s} \in \mathbb{R}^3$ and rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ is derived from the quaternion $\mathbf{q} \in \mathbb{R}^4$. Each Gaussian ellipsoid is also assigned an opacity $\sigma \in \mathbb{R}$ and spherical harmonics (SH) coefficients, which respectively represent the volume density and view-dependent radiance within a nearby local region.

Following Zwicker *et al.* [63], given a world-to-camera rotation $\mathbf{W}$ and the Jacobian $\mathbf{J}$ of the affine approximation of the projective transformation, the EWA splatting algorithm illustrates how to approximately project a 3D Gaussian ellipsoid onto the image plane to determine its effective range and per-pixel opacity values on this image plane. We can obtain the corresponding 2D Gaussian distribution $\mathcal{N}(\hat{\mu}, \hat{\mathbf{\Sigma}})$ as:

$$\hat{\mathbf{\Sigma}} = \mathbf{J}\mathbf{W}\mathbf{\Sigma}\mathbf{W}^T\mathbf{J}^T \ , \tag{2}$$

and $\hat{\mu}$ is the 2D pixel location of a 3D gaussian primitive center.

Fast Rasterization-based Rendering. Fast Gaussian splatting rasterizer enables efficient pixel-by-pixel parallel rendering, and is fully differentiable, which provides a useful GPU-accelerated framework. For an incoming frame, the rasterizer can pre-sort all visible Gaussian primitives in order of depth from near to far. In terms of color rendering, the Gaussian splatting rasterizer adopts an α-blending solution, which accumulates radiance c and opacity values σ on a given pixel by traversing the above depth queue as follows:

$$\hat{I} = \sum_{i=1}^{N} \alpha_i T_i c_i \ , \tag{3}$$

$$T_i = \prod_{k=1}^{i-1} (1 - \alpha_k) \ , \tag{4}$$

$$\alpha_i = \mathcal{N}(\hat{\mu}_\mathbf{i}, \hat{\mathbf{\Sigma}}_\mathbf{i}) \sigma_i \ , \tag{5}$$

where $\hat{I}$ is the rendered color, T_i is the accumulated transmittance, α_i is the opacity contributed to a pixel, and c_i is the color of a Gaussian primitive computed from its SH coefficients. N is the number of Gaussian primitives involved in the splatting process of a pixel. In terms of depth rendering, considering the loss term designed for geometry consistency, our rasterizer provides not only α-blending depth $\hat{D}_{alpha}$ but also the median depth $\hat{D}_{median}$:

$$\hat{D}_{alpha} = \sum_{i=1}^{N} \alpha_i T_i d_i \ , \tag{6}$$

$$\hat{D}_{median} = d_{median} \ , \tag{7}$$

where d_i is the depth of a Gaussian primitive. For a pixel, in its splatting process, we regard a Gaussian at which the cumulative transmittance T falls below $\mathbf{0.5}$ for the first time as the "median Gaussian". Its depth is recorded as d_{median}. T_{median} is the cumulative transmittance at this median Gaussian.

$$(T_{median} \geq 0.5) \ \ and \ \ (T_{median+1} < 0.5) \ . \tag{8}$$

Besides, the accumulated opacity value $\hat{O}$ is similarly required to distinguish unobserved areas for spawning Gaussians:

$$\hat{O} = \sum_{i=1}^{N} \alpha_i T_i \ . \tag{9}$$

3.2 Uncertainty Modeling

Uncertainty model remains a trending topic in multi-view 3D reconstruction in recent decades. Inspired by [38], we believe that explicitly modeling uncertainty in our 3D Gaussian field has a positive effect on increasing the ratio of informative

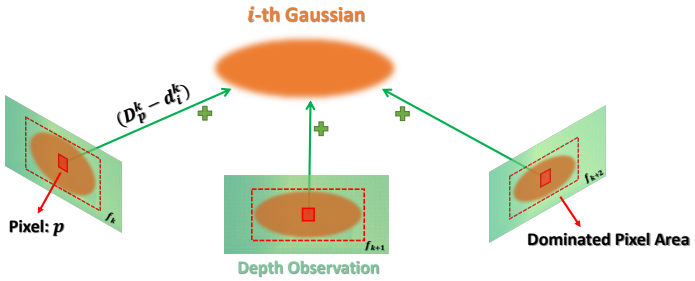

Fig. 3: Uncertainty of the Gaussian Primitives. Uncertainty of a Gaussian primitive is derived from its dominated pixels and corresponding depth biases, reflecting the geometric value and confidence of this primitive.

Gaussians, which is crucial for the robustness and conciseness of a SLAM system. Hence, we propose an uncertainty model suitable for RGB-D observations from two perspectives: rendering images and Gaussian primitives.

Uncertainty Map. α-blending depth is essentially an expected value calculated by sampling Gaussian ellipsoids along a pixel ray. Under the reasonable assumption of a normal distribution, we think that the uncertainty map is highly related to its variance. We can render an uncertainty value from the 3D Gaussian field as in Eq. (10).

$$U = \sum_{i=1}^{N} \alpha_i T_i (\ d_i - D\ )^2\ , \tag{10}$$

where D represents depth observations from the camera sensor. To mitigate drastic changes in positions of Gaussian primitives during optimization, we proposed a geometry variance loss term (Eq. (11)) based on the $H \times W$ uncertainty map to force them to be near the observed depth.

$$\mathcal{L}_{var} = \frac{1}{HW} \sum_{n=1}^{HW} |U_n|\ . \tag{11}$$

Uncertainty of Gaussian primitives. From the perspective of geometric consistency, we design a loss term $\mathcal{L}_{align}$ as in Eq. (12) to align the α-blending depth and median depth.

$$\mathcal{L}_{align} = \frac{1}{HW} \sum_{n=1}^{HW} |\hat{D}^n_{alpha} - \hat{D}^n_{median}|\ . \tag{12}$$

The α-blending depth of a pixel is dominated by the Gaussian primitive with the largest weight. We call this pixel a "dominated pixel" of this maximum weight Gaussian primitive. Our alignment loss forces $\hat{D}_{alpha}$ and $\hat{D}_{median}$ to be similar, which makes this maximum weight Gaussian primitive always occur at d_{median}. Thus, as shown in Eq. (13), the uncertainty v_i of the ith Gaussian primitive

is determined by the difference between its depth and depth observations from all its dominated pixels within a keyframe window $F = \{f_1, f_2, ..., f_k\}$. Fig. 3 further visualizes that a Gaussian primitive receives depth difference values from its dominated pixels in multiple viewpoints.

$$\nu_i = \frac{1}{M_1 + ... + M_k} \sum_{f_k \in F} \sum_{p=1}^{M_k} \alpha_i^{k,p} T_i^{k,p} (\ D_p^k - d_i^k\ )^2 \ . \tag{13}$$

In a keyframe f_k, $\alpha_i^{k,p}$ and $T_i^{k,p}$ represent the opacity and transmittance of the ith Gaussian primitive on a pixel p. D_p^k represents the depth observation on a pixel p in f_k. d_i^k is the depth value of the i-th Gaussian primitive at f_k. $\{M_1, ..., M_k\}$ are the number of dominated pixels of the ith Gaussian primitive in different keyframes. Benefiting from uncertainty modeling, we can regularly detect and remove unreliable Gaussian primitives with high uncertainty exceeding a threshold τ=0.025. Specifically, during the mapping optimization, primitives with $v_i > \tau$ will be manually reduced to a low-opacity level. These low-opacity Gaussian primitives can be optimized again to remove truly irreversible ones, which is a more adaptive and reasonable strategy.

3.3 Mapping

We employ various loss functions to update Gaussian properties, aiming for a consistent and stable Gaussian field. In addition to color and $SSIM$ loss in the original 3D Gaussian splatting, previous experience from NeRF-SLAM works suggests that geometry loss is a necessary part. To overcome anisotropic interference (Arrow-shaped Gaussian primitives), we add a soft scale regularization loss in the mapping process. Note that our system performs the initialization at a slightly higher cost, *i.e.*, more optimization iters.

$$\mathcal{L}_{color} = \frac{1}{HW} \sum_{n=1}^{HW} |\hat{I}_n - I_n| \ , \tag{14}$$

$$\mathcal{L}_{ssim} = SSIM(\hat{I}, I) \ , \tag{15}$$

$$\mathcal{L}_{geo} = \frac{1}{HW} \sum_{n=1}^{HW} |\hat{D}_{alpha}^n - D_n| \ , \tag{16}$$

$$\mathcal{L}_{iso} = \frac{1}{G} \sum_{i \in G} max(\frac{max(\{s_i^x, s_i^y, s_i^z\})}{min(\{s_i^x, s_i^y, s_i^z\})}, \epsilon) - \epsilon \ , \tag{17}$$

where I_n and D_n represent ground-truth color and depth, ϵ=1.0 is a hyperparameter that controls the level of anisotropy, and G represents all visible Gaussians under the current view.

$$\begin{aligned} \mathcal{L}_{mapping} = \omega_1 \mathcal{L}_{color} + \omega_2 \mathcal{L}_{ssim} + \omega_3 \mathcal{L}_{geo} \\ + \omega_4 \mathcal{L}_{align} + \omega_5 \mathcal{L}_{iso} + \omega_6 \mathcal{L}_{var} \ . \end{aligned} \tag{18}$$

These loss functions customized for the SLAM task facilitate faster convergence in mapping and lay a solid foundation for subsequent tracking.

Gaussian Management. In initialization, we densely project Gaussian primitives into 3D space based on depth observations of the first frame. In subsequent mapping, we set an empirical threshold $\psi = 0.5$ to extract unobserved or under-constructed pixels where $\hat{O} < \psi$. Then, we utilize color and depth information on these pixels to spawn fresh Gaussian primitives. In addition, we inherited the original splitting, cloning, and removing strategy for Gaussian densification.

3.4 Tracking

In our system, we have proposed the first comprehensive mathematical theory on derivatives w.r.t. pose in 3D Gaussian splatting framework (refer to supplementary), and empirically discovered that the Lie algebraic representation is more advantageous for camera tracking, especially for the rotation, in a Gaussian field. The camera pose optimization, *i.e.*, rotation and translation $\{\mathfrak{so}(3)|T\}$, mainly includes two parts: sequential tracking and sliding bundle adjustment.

Sequential Tracking. Given the fixed scene representation, the camera pose is initially guessed via the constant speed assumption where the last pose is transformed by the last relative transformation, and then we refine this rough pose using similar photometric and geometric losses weighted by $\lambda = \{\lambda_1, \lambda_2\}$.

$$\mathcal{L}_{tracking} = \lambda_1 \mathcal{L}_{color} + \lambda_2 \mathcal{L}_{geo} \ , \tag{19}$$

$$\{\mathfrak{so}(3)|T\} = \underset{\{\mathfrak{so}(3)|T\}}{argmin}(\mathcal{L}_{tracking}) \ . \tag{20}$$

Sliding Bundle Adjustment. Cumulative error is a typical problem in SLAM, also in Gaussian-based SLAM systems. To ease it, we set up a sliding window F containing k co-visible keyframes and jointly optimize camera extrinsics and scene representation in this window. Due to the efficiency, we encode keyframes into a descriptor pool with a pre-trained NetVLAD [2] model to determine co-visibility through the cosine similarity scores, instead of the view frustum overlap method in previous works. In addition to keyframes from NetVLAD [2], we also added the current frame and the most recent keyframes in the sliding window considering temporal associations. We still employ $\mathcal{L}_{mapping}$ in sliding bundle adjustment, where Ψ is a set of all optimizable Gaussian properties.

$$\Psi, \{\mathfrak{so}(3)|T\} = \underset{\Psi, \{\mathfrak{so}(3)|T\}}{argmin} \ (\mathcal{L}_{mapping}) \ . \tag{21}$$

4 Experiments

In this section, we describe our experimental setup and validate that the proposed system can achieve improvement in both accuracy (Sec. 4.1 and Sec. 4.2) and efficiency (Sec. 4.3). We also confirmed the effectiveness of our design choices

(Sec. 4.4). Additionally, we demonstrate our advantages in image rendering and capability for online third-person view rendering in supplementary. We color each cell as **best**, **second best**, and **third best**.

Datasets. To evaluate our system in various scenarios, we use three standard benchmarks: Replica [43], TUM [44], and ScanNet [10]. The Replica dataset contains 8 available synthetic RGB-D sequences generated by Sucar *et al*. [45]. We examined the generalization of our method on real-world TUM [44] and ScanNet [10] datasets, which contain 5 and 6 challenging scenes respectively.

Implementation Details. We run our system on a desktop equipped with an Intel i9-14900K and an NVIDIA RTX 4090 GPU. We set the learning rate of $\{\mathfrak{so}(3)|T\}$ to $\{0.0015, 0.00215\}$ in sequential tracking in all experiments. For the Replica [43] dataset, we perform 60-iteration mapping with weights $\omega = \{$ 0.7, 0.1, 0.25, 0.25, 0.1, 0.15 $\}$ in a sliding window with $k = 4$ keyframes and 15-iteration sequential pose optimization weighted by $\lambda = \{0.2, 1.0\}$. We select keyframes at an interval of 30. For TUM [44] and ScanNet [10] datasets, we use $\omega = \{$ 1.0, 0.1, 0.8, 0.5, 0.1, 0.5 $\}$, $k = 4$, $\lambda = \{1.0, 0.6\}$. Faced with challenging real-world scenes, we need to extract more keyframes at an interval of 15, perform mapping at 40∼50 iterations, and increase tracking iterations to 25. For further implementation details, please refer to our supplementary.

Metrics. We quantitatively evaluate reconstruction quality using different 3D metrics. Given 3D triangle meshes, we compute mapping *Accuracy* [cm], *Completion* [cm], and *Completion Ratio* [<5cm %]. Following NICE-SLAM [61], we discard unobserved regions that are not in any viewpoints. As for tracking performance, we measure ATE RMSE [44] for estimated trajectories.

Baselines. We primarily consider state-of-the-art NeRF-SLAM works, including NICE-SLAM [61], Co-SLAM [50], Point-SLAM [37], and Vox-Fusion [56], as baselines. For a fair comparison, we reproduced all results from these baselines and reported their reconstruction performance with the same evaluation mechanism. We also add some concurrent manuscripts such as GS-SLAM [54] and SplaTAM [21] for reference, and we directly report the results in their papers.

4.1 Localization Evaluation

We report the localization accuracy of our system in 8 Replica [43] scenes in Tab. 1, where we surpass all other methods by a notable margin of around 25%∼75%. In our Gaussian-based system, image-level pose optimization and the well-designed Gaussian field promote fast and stable convergence to an optimal solution. This is the reason why we have lower variances and higher accuracy. As shown in Tab. 2, despite noisy and sparse depth information in the real-world TUM-RGBD dataset [44], our method still achieves better or competitive performance in 5 selected scenarios. We also benchmark our method and baselines on the similarly challenging ScanNet [10] to compare their tracking performance in Tab. 3. Sensor data from multiple large-scale ScanNet scenes suffers from severe motion blur and specular reflections. Our method further demonstrates its effectiveness and superiority on this dataset, excelling or maintaining com-

Table 1: Tracking Results on the Replica Dataset [43] (ATE RMSE [cm] ↓). Our system consistently achieved the best performance in this dataset, both for 8 individual scenes and for the average. It is worth noting that GS-SLAM [54] and SplaTAM [21] are concurrent with ours.

Method	rm-0	rm-1	rm-2	off-0	off-1	off-2	off-3	off-4	Avg.
NICE-SLAM	0.97	1.31	1.07	0.88	1.00	1.06	1.10	1.13	1.06
Co-SLAM	0.77	1.04	1.09	0.58	0.53	2.05	1.49	0.84	0.99
Point-SLAM	0.56	0.47	0.30	0.35	0.62	0.55	0.72	0.73	0.54
Vox-Fusion	0.40	0.54	0.54	0.50	0.46	0.75	0.50	0.60	0.54
GS-SLAM	0.48	0.53	0.33	0.52	0.41	0.59	0.46	0.70	0.50
SplaTAM	0.31	0.40	0.29	0.47	0.27	0.29	0.32	0.55	0.36
Ours	**0.29**	**0.27**	**0.25**	0.33	**0.14**	**0.28**	**0.31**	**0.29**	**0.27**
Ours-light	0.44	0.36	0.33	**0.29**	0.27	0.43	0.52	0.58	0.40

Table 2: Tracking Results on the TUM-RGBD Dataset [44] (ATE RMSE [cm] ↓). Our system achieves better tracking accuracy and lower variance in different scenarios. "-" indicates unavailable results because the related work is not open source.

Method	fr1/desk	fr1/desk2	fr1/room	fr2/xyz	fr3/office	Avg.
NICE-SLAM	4.26	4.99	34.49	31.73	3.87	15.87
Co-SLAM	2.7	4.57	30.16	1.9	2.6	8.38
Point-SLAM	4.34	**4.54**	30.92	1.31	3.48	8.92
Vox-Fusion	3.52	6.00	19.53	1.49	26.01	11.31
GS-SLAM	3.3	-	-	1.3	6.6	-
SplaTAM	3.35	6.54	11.13	1.24	5.16	5.48
Ours	**2.43**	**4.54**	**9.39**	**1.20**	**2.45**	**4.0**
Ours-light	3.14	4.73	10.67	1.28	2.60	4.48

Table 3: Tracking Results on the ScanNet Dataset [10] (ATE RMSE [cm] ↓). Our method achieves state-of-the-art tracking results in 6 scenes and exceeds other methods on average. "-" indicates failure results in Vox-Fusion [56].

Method	Sc.0000	Sc.0059	Sc.0106	Sc.0169	Sc.0181	Sc.0207	Avg.
NICE-SLAM	12.00	14.00	**7.90**	10.90	13.40	6.20	10.70
Co-SLAM	7.18	12.29	10.9	**6.62**	13.43	7.13	9.37
Point-SLAM	10.24	8.29	11.86	22.16	14.77	9.54	12.19
Vox-Fusion	8.39	8.95	-	9.50	12.20	6.43	-
SplaTAM	12.83	10.10	17.72	12.08	**11.10**	7.46	11.88
Ours	7.09	**7.46**	8.88	8.16	11.60	**5.34**	**8.08**
Ours-light	**5.62**	8.73	9.78	7.93	12.02	5.45	8.17

petitiveness in various scenes. Extensive experiments showcase the remarkable ability of our proposed system to track and handle complex situations.

4.2 Reconstruction Evaluation

In Tab. 4, we quantitatively measure the mapping performance of our proposed system, in comparison to NICE-SLAM [61], Co-SLAM [50], Point-SLAM [37], and Vox-Fusion [56]. It can be observed that our method outperforms all baselines on mapping accuracy. We use TSDF-Fusion [9] to construct our mesh map. We achieve state-of-the-art reconstruction in observed areas. It is worth noting that the Gaussian-based method neither has a global MLP nor a fully covered

Table 4: Reconstruction Results on the Replica [43] Dataset. In terms of mapping accuracy, our method can outperform all existing methods. Due to the inherent limitation of 3D Gaussian representation, our method is slightly worse in completion.

Method	Metric	rm-0	rm-1	rm-2	off-0	off-1	off-2	off-3	off-4	Avg.
NICE-SLAM	Acc.[cm]↓	3.53	3.60	3.03	5.56	3.35	4.71	3.84	3.35	3.87
	Comp.[cm]↓	3.40	3.62	3.27	4.55	4.03	3.94	3.99	4.15	3.87
	Comp.Ratio[<5cm %]↑	86.05	80.75	87.23	79.34	82.13	80.35	80.55	82.88	82.41
Co-SLAM	Acc.[cm]↓	2.11	1.68	1.99	1.57	1.31	2.84	3.06	2.23	2.10
	Comp.[cm]↓	**2.02**	**1.81**	**1.96**	1.56	**1.59**	**2.43**	**2.72**	**2.52**	**2.08**
	Comp.Ratio[<5cm %]↑	**95.26**	**95.19**	**93.58**	**96.09**	**94.65**	**91.63**	**90.72**	**90.44**	**93.44**
Point-SLAM	Acc.[cm]↓	1.45	1.14	1.19	1.05	0.86	1.31	1.57	1.51	1.26
	Comp.[cm]↓	3.46	3.02	2.65	1.65	2.21	3.62	3.47	3.90	3.00
	Comp.Ratio[<5cm %]↑	88.48	89.44	90.13	93.39	90.51	86.17	86.00	85.74	88.73
Vox-Fusion	Acc.[cm]↓	1.77	1.51	2.23	1.63	1.44	2.09	2.33	2.02	1.88
	Comp.[cm]↓	2.69	2.31	2.58	1.87	1.66	3.03	2.81	3.51	2.56
	Comp.Ratio[<5cm %]↑	92.03	92.47	90.13	93.86	94.40	88.94	89.10	86.53	90.93
Ours	Acc.[cm]↓	**1.10**	**0.97**	**0.96**	**0.85**	**0.67**	**1.10**	**1.28**	**1.16**	**1.01**
	Comp.[cm]↓	3.26	2.77	2.79	**1.49**	2.15	3.34	3.23	3.66	2.84
	Comp.Ratio[<5cm %]↑	88.26	89.48	89.10	93.60	90.14	86.04	85.78	85.66	88.51
Ours-light	Acc.[cm]↓	1.17	0.98	0.99	0.87	0.71	1.2	1.36	1.26	1.06
	Comp.[cm]↓	3.32	2.65	2.81	1.51	2.20	3.44	3.27	3.78	2.87
	Comp.Ratio[<5cm %]↑	88.20	89.33	89.10	93.41	90.14	85.64	85.47	85.30	88.43

feature grid, as in Co-SLAM [50]. Consequently, such a system exhibits a slightly weaker hole-filling ability compared to the NeRF-based method, which explains why our system is slightly worse in the *Completion* metric. As shown in Fig. 4, we qualitatively present the ground truth mesh and mesh reconstructions from ours and other baselines. Evidently, our system achieves more detailed geometry and less noisy outliers.

4.3 Runtime and Memory Analysis

We evaluate the runtime and memory footprint of our system compared to other works in Tab. 5. We reported the tracking and mapping efficiency in terms of per-iteration time consumption and the total number of optimization iterations. With the support of the GPU-accelerated rasterizer, our system can operate at around 8.5Hz. Meanwhile, our carefully designed pipeline allows this system to expand to a lightweight version and to work with half-resolution images. For tracking, this lightweight version can work twice as fast as the original one, at the cost of a slight decrease in accuracy, as shown in Tabs. 1 to 3. For mapping, our lightweight version demonstrates similar superiority in Tab. 4. It can be clearly seen that both our versions achieve better performance than Co-SLAM [50]. Further efficiency analysis on TUM-RGBD [44] and ScanNet [10] is shown in supplementary. Also, our customized Gaussian field allows us to outperform the concurrent Gaussian-based works with less computational burden.

However, as a non-MLP scene representation, the 3D Gaussian field inevitably requires much memory consumption to store different properties. This reason results in a considerable memory footprint in the Gaussian-based SLAM system, which is a common limitation in other Gaussian-based research topics.

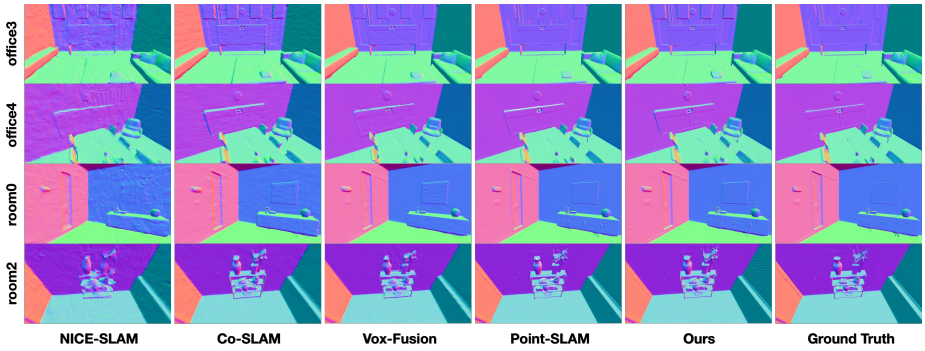

Fig. 4: Reconstruction Performance on Replica [43] Dataset. We qualitatively compared the mesh reconstruction results from CG-SLAM and other baselines, where CG-SLAM can produce more detailed geometry at a lower computation cost.

Table 5: Runtime and Memory Usage. We comprehensively compared the runtime and memory usage on Replica [43] Office 0. Our proposed CG-SLAM can perform more efficient tracking and mapping than existing works, actually reaching a real-time level. "-" indicates unavailable results in related works.

Method	Tracking [ms× it]↓	Mapping [ms× it]↓	Mapping Interval	System FPS↑	Decoder Param↓	Scene Embeeding↓
Vox-Fusion	23.61 × 30	86.55 × 10	10	1.25	0.98 MB	0.162 MB
NICE-SLAM	6.19 × 10	91.59 × 60	5	0.86	0.43 MB	89.56 MB
Co-SLAM	4.45 × 10	11.9 × 10	5	14.64	6.37 MB	-
Point-SLAM	6.14 × 40	22.25× 300	5	0.63	0.54 MB	28.11 MB
GS-SLAM	11.9 × 10	12.8× 100	-	8.34	-	-
SplaTAM	41.7 × 40	50.1 × 60	1	0.21	-	-
Ours	6.02 × 10	28.9 × 60	30	8.47	-	231.66 MB
Ours-light	3.80 × 10	10.70 × 60	30	**16.83**	-	56.50 MB

4.4 Ablation Study

To verify the rationality of our designs, we investigate the effectiveness of the anisotropy regularization, alignment and variance losses, and uncertainty model.

Effectiveness of Isotropy Loss. Ablation results in Tab. 6 reveal how the anisotropy regularization term affects tracking metrics. We can notice that anisotropy (w/o $\mathcal{L}_{iso}$) will reduce the tracking accuracy by disrupting the view extrapolation of a 3D Gaussian field. Oppositely, excessive regularization (Isotropic Gaussians) will degrade rendering and subsequently affect the photometric loss, leading to poor tracking performance. Overall, a soft controllable constraint like $\mathcal{L}_{iso}$ is a reasonable choice. To more intuitively illustrate this phenomenon, we display opacity maps with and without anisotropy regularization in Fig. 5 (b), and apparent arrow-shaped artifacts occur in the latter.

Effectiveness of Alignment and Variance Losses. Alignment and variance losses push primitives closer to object surfaces, facilitating novel view syn-

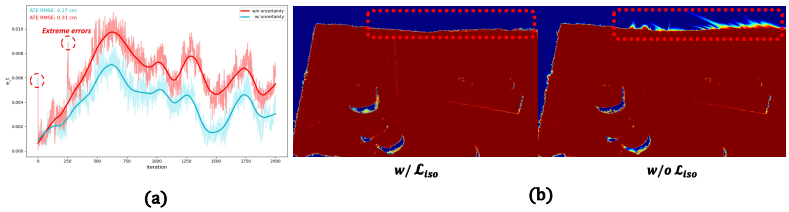

Fig. 5: Uncertainty Model Ablation and Anisotropy Interference. (a) Uncertainty Model Ablation. This plot illustrates that the uncertainty model helps improve tracking accuracy while avoiding some extreme errors. (b) Anisotropy Interference. It can be clearly seen that in the case of w/o $\mathcal{L}_{iso}$, serious arrow-shaped artifacts occur on the edges of the image.

Table 6: Isotropy Loss Ablation Results(ATE RMSE [cm] ↓). The experimental results demonstrate the effectiveness of our anisotropy regularization term. "-" indicates a failure situation.

Setting	rm-0	rm-1	rm-2	off-0	off-1	off-2	off-3	off-4	Avg.
w/ $\mathcal{L}_{iso}$	**0.29**	**0.27**	**0.25**	0.33	**0.14**	**0.28**	**0.31**	**0.29**	**0.27**
w/o $\mathcal{L}_{iso}$	0.32	0.31	0.54	0.36	0.24	**0.31**	0.72	-	-
Isotropic Gaussians	0.30	0.32	1.04	**0.29**	7.62	0.38	1.54	1.36	1.61

thesis and SLAM. The effectiveness of these two losses is demonstrated in the supplementary.

Effectiveness of Uncertainty Model. As shown in Fig. 5 (a), tracking error curves show that our uncertainty model can improve tracking accuracy while avoiding some extreme errors. For further quantitative ablation results, refer to the supplementary material.

5 Conclusion

We have proposed CG-SLAM, a dense RGB-D SLAM based on a consistent and uncertainty-aware 3D Gaussian field. Our targeted loss functions strengthen the 3D Gaussian field in terms of consistency and stability. The uncertainty model further distills highly informative primitives in this field to reduce interference from outliers. Besides, a customized GPU-accelerated rasterization pipeline enables our system to achieve state-of-the-art accuracy and efficiency in various scenes. Through extensive experiments, it can be concluded that our method outperforms previous works regarding tracking, mapping, and efficiency.

Limitations. Considerable memory usage is one limitation of the Gaussian-based system. We expect that a more compact Gaussian field can be adopted in the SLAM task. In addition, the Gaussian-based method has a weak prediction ability for unobserved areas. Moreover, our system is not capable of handling dynamic objects in the environment. We believe it is a very interesting direction for future work.

Acknowledgements

This work was partially supported by the NSFC (No. 62102356).

References

1. Abdal, R., Yifan, W., Shi, Z., Xu, Y., Po, R., Kuang, Z., Chen, Q., Yeung, D.Y., Wetzstein, G.: Gaussian shell maps for efficient 3d human generation (2023)
2. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5297–5307 (2016)
3. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields (2021)
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR (2022)
5. Chen, H., Li, C., Lee, G.H.: Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance (2023)
6. Chen, J., Bautembach, D., Izadi, S.: Scalable real-time volumetric surface reconstruction. ACM Trans. Graph. **32**(4), 113:1–113:16 (2013). https://doi.org/10.1145/2461912.2461940, https://doi.org/10.1145/2461912.2461940
7. Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., Lin, G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting (2023)
8. Chung, C., Tseng, Y., Hsu, Y., Shi, X.Q., Hua, Y., Yeh, J., Chen, W., Chen, Y., Hsu, W.H.: Orbeez-slam: A real-time monocular visual SLAM with ORB features and nerf-realized mapping. In: IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023. pp. 9400–9406. IEEE (2023). https://doi.org/10.1109/ICRA48891.2023.10160950, https://doi.org/10.1109/ICRA48891.2023.10160950
9. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. pp. 303–312 (1996)
10. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
11. Duisterhof, B.P., Mandi, Z., Yao, Y., Liu, J.W., Shou, M.Z., Song, S., Ichnowski, J.: Md-splatting: Learning metric deformation from 4d gaussians in highly deformable scenes (2023)
12. Fang, J., Wang, J., Zhang, X., Xie, L., Tian, Q.: Gaussianeditor: Editing 3d gaussians delicately with text instructions. arXiv preprint arXiv:2311.16037 (2023)
13. Fang, J., Yi, T., Wang, X., Xie, L., Zhang, X., Liu, W., Nießner, M., Tian, Q.: Fast dynamic radiance fields with time-aware neural voxels. In: SIGGRAPH Asia 2022 Conference Papers (2022)
14. Fridovich-Keil and Yu, Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022)
15. Geng, C., Peng, S., Xu, Z., Bao, H., Zhou, X.: Learning neural volumetric representations of dynamic humans in minutes. In: CVPR (2023)

16. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. arXiv preprint arXiv:2311.12775 (2023)
17. Hu, J., Mao, M., Bao, H., Zhang, G., Cui, Z.: Cp-slam: Collaborative neural point-based slam system. Advances in Neural Information Processing Systems **36** (2024)
18. Irshad, M.Z., Zakharov, S., Liu, K., Guizilini, V., Kollar, T., Gaidon, A., Kira, Z., Ambrus, R.: Neo 360: Neural fields for sparse view synthesis of outdoor scenes (2023), https://arxiv.org/abs/2308.12967
19. Jena, R., Iyer, G.S., Choudhary, S., Smith, B., Chaudhari, P., Gee, J.: Splatarmor: Articulated gaussian splatting for animatable humans from monocular rgb videos. arXiv preprint arXiv:2311.10812 (2023)
20. Katragadda, S., Lee, W., Peng, Y., Geneva, P., Chen, C., Guo, C., Li, M., Huang, G.: Nerf-vins: A real-time neural radiance field map-based visual-inertial navigation system. CoRR **abs/2309.09295** (2023). https://doi.org/10.48550/ARXIV.2309.09295, https://doi.org/10.48550/arXiv.2309.09295
21. 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)
22. 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, Seattle, Washington, USA, June 29 - July 1, 2013. pp. 1–8. IEEE Computer Society (2013). https://doi.org/10.1109/3DV.2013.9, https://doi.org/10.1109/3DV.2013.9
23. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)
24. Kerl, C., Sturm, J., Cremers, D.: Dense visual slam for rgb-d cameras. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 2100–2106. IEEE (2013)
25. Koestler, L., Yang, N., Zeller, N., Cremers, D.: Tandem: Tracking and dense mapping in real-time using deep multi-view stereo. In: Conference on Robot Learning (CoRL) (2021)
26. Li, Y., Liu, X., Dong, W., Zhou, H., Bao, H., Zhang, G., Zhang, Y., Cui, Z.: Deltar: Depth estimation from a light-weight tof sensor and rgb image. In: European conference on computer vision. pp. 619–636. Springer (2022)
27. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: IEEE International Conference on Computer Vision (ICCV) (2021)
28. Liu, X., Li, Y., Teng, Y., Bao, H., Zhang, G., Zhang, Y., Cui, Z.: Multi-modal neural radiance field for monocular dense slam with a light-weight tof sensor. In: Proceedings of the ieee/cvf international conference on computer vision. pp. 1–11 (2023)
29. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: The European Conference on Computer Vision (ECCV) (2020)
30. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). https://doi.org/10.1145/3528223.3530127, https://doi.org/10.1145/3528223.3530127
31. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.W.: Kinectfusion: Real-time dense surface mapping and tracking. In: 10th IEEE International Symposium on Mixed

and Augmented Reality, ISMAR 2011, Basel, Switzerland, October 26-29, 2011. pp. 127–136. IEEE Computer Society (2011). https://doi.org/10.1109/ISMAR.2011.6092378, https://doi.org/10.1109/ISMAR.2011.6092378
32. Newcombe, R.A., Lovegrove, S., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: Metaxas, D.N., Quan, L., Sanfeliu, A., Gool, L.V. (eds.) IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011. pp. 2320–2327. IEEE Computer Society (2011). https://doi.org/10.1109/ICCV.2011.6126513, https://doi.org/10.1109/ICCV.2011.6126513
33. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. ACM Trans. Graph. **32**(6), 169:1–169:11 (2013). https://doi.org/10.1145/2508363.2508374, https://doi.org/10.1145/2508363.2508374
34. Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., Nieto, J.I.: Voxblox: Incremental 3d euclidean signed distance fields for on-board MAV planning. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017. pp. 1366–1373. IEEE (2017). https://doi.org/10.1109/IROS.2017.8202315, https://doi.org/10.1109/IROS.2017.8202315
35. Pang, H., Zhu, H., Kortylewski, A., Theobalt, C., Habermann, M.: Ash: Animatable gaussian splats for efficient and photoreal human rendering (2023)
36. Rosinol, A., Leonard, J.J., Carlone, L.: Nerf-slam: Real-time dense monocular SLAM with neural radiance fields. In: IROS. pp. 3437–3444 (2023). https://doi.org/10.1109/IROS55552.2023.10341922, https://doi.org/10.1109/IROS55552.2023.10341922
37. 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)
38. Sandström, E., Ta, K., Van Gool, L., Oswald, M.R.: Uncle-slam: Uncertainty learning for dense neural slam. arXiv preprint arXiv:2306.11048 (2023)
39. 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 (CVPR) (June 2019)
40. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
41. Shao, R., Zheng, Z., Tu, H., Liu, B., Zhang, H., Liu, Y.: Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2023)
42. Steinbrücker, F., Kerl, C., Cremers, D.: Large-scale multi-resolution surface reconstruction from RGB-D sequences. In: IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013. pp. 3264–3271. IEEE Computer Society (2013). https://doi.org/10.1109/ICCV.2013.405, https://doi.org/10.1109/ICCV.2013.405
43. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The replica dataset: A digital replica of indoor spaces (2019)

44. 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)
45. 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)
46. Tateno, K., Tombari, F., Laina, I., Navab, N.: CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 6565–6574. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.695, https://doi.org/10.1109/CVPR.2017.695
47. Teed, Z., Deng, J.: DROID-SLAM: deep visual SLAM for monocular, stereo, and RGB-D cameras. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual. pp. 16558–16569 (2021), https://proceedings.neurips.cc/paper/2021/hash/89fcd07f20b6785b92134bd6c1d0fa42-Abstract.html
48. Tretschk, E., Golyanik, V., Zollhöfer, M., Bozic, A., Lassner, C., Theobalt, C.: Scenerflow: Time-consistent reconstruction of general dynamic scenes. In: International Conference on 3D Vision (3DV) (2024)
49. Vespa, E., Nikolov, N., Grimm, M., Nardi, L., Kelly, P.H.J., Leutenegger, S.: Efficient octree-based volumetric SLAM supporting signed-distance and occupancy mapping. IEEE Robotics Autom. Lett. **3**(2), 1144–1151 (2018). https://doi.org/10.1109/LRA.2018.2792537, https://doi.org/10.1109/LRA.2018.2792537
50. 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)
51. Wang, T., Zhang, B., Zhang, T., Gu, S., Bao, J., Baltrusaitis, T., Shen, J., Chen, D., Wen, F., Chen, Q., et al.: Rodin: A generative model for sculpting 3d digital avatars using diffusion. arXiv preprint arXiv:2212.06135 (2022)
52. Whelan, T., Salas-Moreno, R.F., Glocker, B., Davison, A.J., Leutenegger, S.: Elasticfusion: Real-time dense SLAM and light source estimation. Int. J. Robotics Res. **35**(14), 1697–1716 (2016). https://doi.org/10.1177/0278364916669237, https://doi.org/10.1177/0278364916669237
53. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. arXiv preprint arXiv:2201.08845 (2022)
54. Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19595–19604 (2024)
55. Yang, N., von Stumberg, L., Wang, R., Cremers, D.: D3VO: deep depth, deep pose and deep uncertainty for monocular visual odometry. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 1278–1289. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.00136, https://openaccess.thecvf.com/content_CVPR_2020/html/Yang_D3VO_Deep_Depth_Deep_Pose_and_Deep_Uncertainty_for_Monocular_CVPR_2020_paper.html
56. Yang, X., Li, H., Zhai, H., Ming, Y., Liu, Y., Zhang, G.: Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In: 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 499–507. IEEE (2022)

57. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. arXiv preprint arXiv:2312.00732 (2023)
58. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelNeRF: Neural radiance fields from one or few images. In: CVPR (2021)
59. Yuan, Y., Li, X., Huang, Y., De Mello, S., Nagano, K., Kautz, J., Iqbal, U.: Gavatar: Animatable 3d gaussian avatars with implicit mesh learning. arXiv preprint arXiv:2312.11461 (2023)
60. Zhou, S., Chang, H., Jiang, S., Fan, Z., Zhu, Z., Xu, D., Chari, P., You, S., Wang, Z., Kadambi, A.: Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. arXiv preprint arXiv:2312.03203 (2023)
61. 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)
62. Zielonka, W., Bagautdinov, T., Saito, S., Zollhöfer, M., Thies, J., Romero, J.: Drivable 3d gaussian avatars (2023)
63. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Ewa volume splatting. In: Proceedings Visualization, 2001. VIS'01. pp. 29–538. IEEE (2001)