

# VIGS-SLAM: Visual Inertial Gaussian Splatting SLAM

Zihan Zhu<sup>1</sup>, Wei Zhang<sup>2</sup>, Moyang Li<sup>1</sup>, Norbert Haala<sup>2</sup>,  
Marc Pollefeys<sup>1,3</sup>, and Daniel Barath<sup>1</sup>

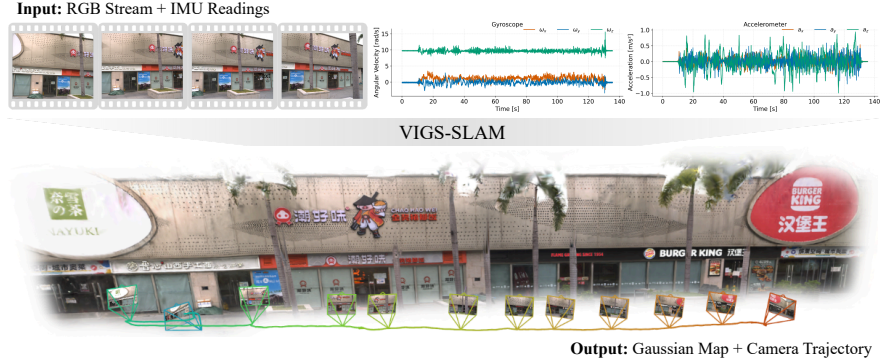
<sup>1</sup> ETH Zurich, Zurich, Switzerland

<sup>2</sup> University of Stuttgart, Stuttgart, Germany

<sup>3</sup> Microsoft

**Abstract.** We present VIGS-SLAM, a visual-inertial 3D Gaussian Splatting SLAM system that achieves robust real-time tracking and high-fidelity reconstruction. Although recent 3DGS-based SLAM methods achieve dense and photorealistic mapping, their purely visual design degrades under challenging conditions such as motion blur, low texture, and exposure variations. Our method tightly couples visual and inertial cues within a unified optimization framework, jointly optimizing camera poses, depths, and IMU states. It features robust IMU initialization, time-varying bias modeling, and loop closure with consistent Gaussian updates. Experiments on five challenging datasets demonstrate our superiority over state-of-the-art methods. The code will be made public.

**Keywords:** SLAM · IMU · Gaussian Splatting



**Fig. 1: VIGS-SLAM.** Given a sequence of RGB frames and IMU readings, our method robustly tracks the camera trajectory while reconstructing a high-quality Gaussian map. Above is the visualization of *Retail* sequence in FAST-LIVO2 [65] dataset.

## 1 Introduction

Simultaneous Localization and Mapping (SLAM) is a key problem in robotics and computer vision, enabling autonomous navigation, augmented (AR) and mixed reality (VR) applications. Recent advances in neural implicit representations [42] and 3D Gaussian Splatting (3DGS) [27] have transformed SLAM

mapping from sparse point clouds to dense, photorealistic scene reconstructions. In particular, 3DGS SLAM methods [50,62] demonstrate that 3D Gaussian primitives provide a compact yet expressive scene representation, enabling real-time dense mapping and photorealistic novel view synthesis. However, most existing 3DGS-based SLAM systems remain purely visual: they rely on visual correspondences for tracking and often degrade under motion blur, textureless regions, low frame rate, or transient occlusions – conditions common in real-world scenarios.

Meanwhile, inertial measurement units (IMUs) have become ubiquitous – integrated in virtually every modern smartphone [37, 49], AR headset [20, 41], and consumer camera [21–23]. These sensors are low-cost (MEMS often under \$1 per chip) yet provide high-frequency measurements of acceleration and angular velocity, complementing vision by stabilizing tracking, recovering metric scale, and maintaining robustness under visually degraded conditions.

Traditional visual-inertial odometry (VIO) and SLAM such as OKVIS [30], MSCKF [43], VINS-Mono [46], and ORB-SLAM3 [4] are typically formulated as either filtering-based or optimization-based frameworks, that fuse visual observations with inertial measurements. Most systems rely on sparse geometric features such as BRISK [29] or ORB [48], while others adopt direct photometric alignment such as VI-DSO [55] to jointly optimize image intensities. Despite their accuracy, these representations produce only sparse or semi-dense point maps. The recent state-of-the-art visual SLAM system DROID-SLAM [54] reformulates dense correspondence estimation and bundle adjustment within a learned, differentiable framework, enabling iterative refinement of correspondences and joint optimization over camera poses and per-pixel disparities. This formulation is particularly well-suited for Gaussian initialization, as the 3D Gaussians can be directly initialized from the unprojected point cloud derived from the estimated disparities. However, DROID-SLAM [54] and subsequent Gaussian-Splatting extensions [50,62] remain purely visual, without exploiting inertial measurements that could further enhance robustness.

The very few existing VIO 3DGS-SLAM approaches [44,53,58] exhibit several limitations, such as reliance on depth sensors, naive fusion of IMU data through pairwise (non-windowed) constraints, fixed IMU bias, or decoupled alternating optimization. Therefore, they suffer from reduced accuracy, limited robustness, and blurry renderings. In contrast, we develop **VIGS-SLAM**, a highly robust SLAM system that achieves both *real-time accurate tracking* and *high-fidelity Gaussian reconstruction*. Our method tightly couples visual and inertial energy terms into a unified optimization framework, incorporates robust multi-stage IMU initialization and efficient loop closure with consistent Gaussian updates. This design improves tracking accuracy, reduces drift, enhances robustness under challenging conditions (e.g., motion blur, low texture, exposure variations, dynamic objects), and yields higher-fidelity renderings. Extensive experiments on five diverse datasets, including a challenging self-captured visual-inertial dataset, demonstrate consistent improvements over state-of-the-art methods.

## 2 Related Work

**Visual-Inertial Odometry and SLAM.** An early and seminal visual-inertial odometry approach is MSCKF [43], which introduced a feature-marginalizing EKF that maintains a sliding window of cloned states, achieving real-time operation with a bounded state size. Follow-ups [18, 34] improved consistency, observability, and robustness, and inspired open implementations [2, 14].

In parallel, optimization-based methods jointly minimize reprojection and preintegrated IMU residuals via a sliding-window nonlinear least-squares formulation. OKVIS [30] tightly couples reprojection and IMU residuals in a keyframe bundle-adjustment backend. VINS-Mono [46] adds robust initialization, loop closure, and relocalization. ICE-BA [35] streamlines efficient incremental BA. ORB-SLAM3 [4] unifies visual and visual-inertial modes with multi-map management and loop closing. Despite strong accuracy, these pipelines typically rely on sparse features or direct pixel intensities, which makes them brittle under low texture, motion blur, repetitive patterns, strong illumination/exposure changes. They also produce sparse (or semi-dense) maps that limit downstream tasks such as dense reconstruction, semantics, and photorealistic rendering.

DBA-Fusion [67] is the first to leverage dense correspondences from DROID-SLAM [54] for visual-inertial SLAM. Our VIGS-SLAM also utilizes the same dense correspondences, but incorporates and optimizes visual-inertial constraints in a fundamentally different manner. DBA-Fusion [67] employs a cascaded two-stage framework: visual bundle adjustment is first solved independently, and the resulting Schur-complement Hessian is injected as a linearized factor into GTSAM [9] inertial optimizer. This separation introduces a linearization gap that can only be partially mitigated through outer re-linearization iterations, resulting in limited robustness due to the lack of coherent visual-inertial fusion. In contrast, VIGS-SLAM implements custom CUDA kernels for inertial bundle adjustment that directly fuse visual and inertial Hessian contributions within a single normal-equation solve per iteration. This formulation enables true tightly coupled joint optimization over poses, velocities, and biases, leading to improved numerical consistency and robustness.

Furthermore, beyond tightly coupled optimization, our system integrates robust staged IMU initialization, efficient loop closure and Gaussian Splatting-based dense mapping, resulting in a unified visual-inertial Gaussian Splatting SLAM framework that improves tracking accuracy and robustness while enabling high-fidelity dense reconstruction and novel view synthesis.

**Neural Implicit and 3DGS SLAM.** Neural implicit representations and 3D Gaussian Splatting (3DGS) have gained significant attention in SLAM for accurate dense reconstruction and realistic novel view synthesis. Early systems such as iMAP [52] and NICE-SLAM [71] established unified pipelines that jointly perform mapping and tracking with neural fields. Subsequent work improves efficiency and scalability through compact encodings [24, 28, 56], extends to monocular settings [1, 8, 47, 63, 70], and integrates semantics [31, 61, 69]. 3DGS [27] provides an explicit and efficient alternative, which sparked a wave of SLAM sys-





In contrast to prior work, we jointly optimize visual and inertial terms, incorporate robust IMU initialization, and perform efficient loop closure with consistent Gaussian map updates, resulting in improved accuracy and robustness.

### 3 Method

We provide an overview of our pipeline in Fig. 2. Our system takes as input a sequence of RGB frames  $\{I_i\}_{i=1}^N$  and raw IMU measurements  $\{\mathbf{a}_k, \boldsymbol{\omega}_k\}_{k=1}^M$ , where  $\mathbf{a}_k$  and  $\boldsymbol{\omega}_k$  denote the angular velocity and linear acceleration. It simultaneously performs camera tracking and Gaussian Splatting Mapping. In detail, we first perform a staged IMU initialization to recover the metric scale, gravity direction, and IMU parameters. After initialization, incoming frames are processed by the frontend tracking module, which selects keyframes and forms a local frame graph  $\mathcal{E}$ . Camera motion as well as IMU parameters are then optimized within this graph by jointly minimizing both visual and inertial residuals. Parallel to tracking, we incrementally build and refine a 3D Gaussian Splatting Map for high-quality rendering. Loop closures are detected and performed efficiently, with corresponding Gaussian updates to ensure global consistency.

#### 3.1 Tracking

The tracking module processes sequential images online, creating a new keyframe when the optical-flow magnitude to the last keyframe exceeds a threshold. To limit drift from IMU pre-integration over long time intervals, we also force a new keyframe at least every  $t_{kf}$  seconds. We formalize our tracking as an optimization problem where, for each keyframe  $i$ , we optimize camera pose  $\mathbf{T}_i = (\mathbf{R}_i, \mathbf{p}_i)$ , disparity  $\mathbf{d}_i$ , velocity  $\mathbf{v}_i$ , and IMU bias  $\mathbf{b}_i = [\mathbf{b}_{g_i}^T, \mathbf{b}_{a_i}^T]^T$  (gyroscope  $\mathbf{b}_{g_i}$  and accelerometer bias  $\mathbf{b}_{a_i}$ ). In the following, we first introduce the visual and inertial residuals, followed by the optimization modules that build upon these residuals.

**Vision Residual.** Following DROID-SLAM [54], we utilize a learned GRU-based update operator to predict and iteratively update the correspondence  $\mathbf{u}_{ij}^*$  and its associated confidence map  $\mathbf{w}_{ij}$  between a keyframe pair  $(i, j)$ . We jointly optimize per-keyframe camera pose  $\mathbf{T}$  and disparity  $\mathbf{d}$  by minimizing a reprojection error between the refined correspondence  $\mathbf{u}_{ij}^*$  and that computed from the current pose and disparity. The vision residuals are formulated as:

$$E_{\text{vis}}(\mathbf{T}, \mathbf{d}) = \sum_{(i,j) \in \mathcal{E}} \|\mathbf{u}_{ij}^* - \Pi(\mathbf{T}_{ij} \Pi^{-1}(\mathbf{u}_i, \mathbf{d}_i))\|_{\boldsymbol{\Sigma}_{ij}}^2, \quad (1)$$

where  $\boldsymbol{\Sigma}_{ij} = \text{diag}(\mathbf{w}_{ij}^{\text{vis}})$ . Here, we loop over edges  $(i, j)$  within a frame graph  $\mathcal{E}$  connecting co-visible keyframes, and  $\mathbf{T}_{ij}$  represents the relative transform between frames  $i$  and  $j$ . We denote the projection and back-projection functions as  $\Pi$  and  $\Pi^{-1}$ . Parameter  $\mathbf{u}_i$  is the 2D image coordinate in frame  $i$ . Function  $\|\cdot\|_{\boldsymbol{\Sigma}}$  denotes the Mahalanobis distance, weighting the residuals according to the confidence  $\mathbf{w}_{ij}^{\text{vis}}$ .

**Inertial Residual.** For each consecutive keyframe pair  $(i, j)$ , we first perform IMU pre-integration [12] between timestamps  $t_i$  and  $t_j$  to efficiently fuse the high-frequency IMU data. The pre-integration yields relative rotation  $\Delta \mathbf{R}_{ij}$ , position  $\Delta \mathbf{p}_{ij}$ , and velocity  $\Delta \mathbf{v}_{ij}$ , together with the corresponding Jacobians  $(\mathbf{J}_{ij}^{\text{rot}}, \mathbf{J}_{ij}^{\text{pos}}, \mathbf{J}_{ij}^{\text{vel}})$  and covariance  $\Sigma_{ij}^{\text{iner}}$ .

In the inertial residual, we jointly optimize camera pose  $\mathbf{T}_i$ , velocity  $\mathbf{v}_i$  and IMU bias  $\mathbf{b}_i$  by minimizing the discrepancy between the current relative camera motion and the corresponding preintegrated IMU measurements. Additionally, we include a temporal bias smoothness term. The inertial residuals are formulated as follows:

$$E_{\text{iner}}(\mathbf{T}, \mathbf{v}, \mathbf{b}) = \sum_{(i,j=i+1) \in \mathcal{E}} \left\| [(\mathbf{r}_{i,j}^{\text{rot}})^\top, (\mathbf{r}_{i,j}^{\text{pos}})^\top, (\mathbf{r}_{i,j}^{\text{vel}})^\top, (\mathbf{r}_{i,j}^{\text{bias}})^\top]^\top \right\|_{\Sigma_{ij}^{\text{iner}}}^2. \quad (2)$$

Here, the residual terms are defined as:

$$\mathbf{r}_{ij} = \begin{bmatrix} \mathbf{r}_{i,j}^{\text{rot}} \\ \mathbf{r}_{i,j}^{\text{pos}} \\ \mathbf{r}_{i,j}^{\text{vel}} \\ \mathbf{r}_{i,j}^{\text{bias}} \end{bmatrix} = \begin{bmatrix} \text{Log} \left( (\Delta \mathbf{R}_{ij} \text{Exp}(\mathbf{J}_{ij}^{\text{rot}}(\mathbf{b}_{g_i} - \hat{\mathbf{b}}_{g_i})))^\top \mathbf{R}_i^\top \mathbf{R}_j \right) \\ \mathbf{R}_i^\top (\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t_{ij} - \frac{1}{2} \mathbf{g} \Delta t_{ij}^2) - (\Delta \mathbf{p}_{ij} + \mathbf{J}_{ij}^{\text{pos}}(\mathbf{b}_i - \hat{\mathbf{b}}_i)) \\ \mathbf{R}_i^\top (\mathbf{v}_j - \mathbf{v}_i - \mathbf{g} \Delta t_{ij}) - (\Delta \mathbf{v}_{ij} + \mathbf{J}_{ij}^{\text{vel}}(\mathbf{b}_{g_i} - \hat{\mathbf{b}}_{g_i})) \\ \mathbf{b}_j - \mathbf{b}_i \end{bmatrix}, \quad (3)$$

where  $\text{Log}(\cdot)$  and  $\text{Exp}(\cdot)$  are the Lie algebra logarithm and exponential maps. Parameter  $\hat{\mathbf{b}}$  denotes the initial IMU bias and  $\mathbf{g} = \mathbf{R}_{\text{wg}} \mathbf{g}_I$  is the gravity direction in the world frame, where  $\mathbf{g}_I = (0, 0, G)^\top$  denotes the gravity in a gravity-aligned inertial frame,  $G$  denotes its magnitude, and  $\mathbf{R}_{\text{wg}} \in \text{SO}(3)$  is the rotation from the inertial frame to the world frame. For simplicity, we omit the transformation  $T_{cb}$  between the camera and IMU coordinate frames in the equations; however, it is taken into account in the implementation.

**Frontend Tracking: Local Bundle Adjustment.** We maintain a sliding-window local frame graph  $\mathcal{E}$ , following DROID-SLAM [54]. For pose initialization, we use IMU pre-integration to initialize each new keyframe. To prevent unreliable inertial cues, we fall back to the previous-pose initialization whenever the pre-integration uncertainty is high, i.e., when the covariance trace exceeds a predefined threshold,  $\text{tr}(\Sigma_{ij}^{\text{iner}}) > \tau_{\Sigma}^{\text{init}}$ . Using the estimated pose of the new keyframe, we initialize correspondences to earlier frames via geometric warping, providing a stronger starting point for the GRU-based update operator. We then perform joint visual-inertial optimization by minimizing the sum of the vision Eq. (1) and inertial Eq. (2) residuals on the local frame graph  $\mathcal{E}$ . The pose  $\mathbf{T}$ , velocity  $\mathbf{v}$ , bias  $\mathbf{b}$  and disparity  $\mathbf{d}$  are jointly optimized using the Levenberg-Marquardt algorithm through custom CUDA kernels as follows:

$$\begin{bmatrix} \mathbf{B} & \mathbf{E} \\ \mathbf{E}^\top & \mathbf{C} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{\xi} \\ \Delta \mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_\xi \\ \mathbf{w}_d \end{bmatrix} \quad \begin{aligned} \Delta \boldsymbol{\xi} &= [\mathbf{B} - \mathbf{E} \mathbf{C}^{-1} \mathbf{E}^\top]^{-1} (\mathbf{w}_\xi - \mathbf{E} \mathbf{C}^{-1} \mathbf{w}_d) \\ \Delta \mathbf{d} &= \mathbf{C}^{-1} (\mathbf{w}_d - \mathbf{E}^\top \Delta \boldsymbol{\xi}) \end{aligned} \quad (4)$$

where  $\Delta \boldsymbol{\xi}$  represents the update of  $[\mathbf{T}, \mathbf{v}, \mathbf{b}]$ , and  $\Delta \mathbf{d}$  denotes the depth update. Matrix  $\mathbf{C}$  is diagonal as each term in Eq. (1) depends only on a single depth value, thus it can be inverted by  $\mathbf{C}^{-1} = 1/\mathbf{C}$ .

**Loop Closure: Pose Graph Bundle Adjustment (PGBA).** To handle loop closure efficiently, inspired by [62], we adopt PGBA rather than a full global BA, trading a small amount of accuracy for substantial speed up. A parallel loop-detector builds loop edges  $\mathcal{E}^*$  based on optical flow differences. We dynamically grow the relative pose graph  $\mathcal{E}^+$  by adding the relative pose from local frame graph  $\mathcal{E}$ . We restrict heavy vision updates to only loop closure pairs  $\mathcal{E}^*$  and add lightweight relative-pose constraints over the whole pose graph  $\mathcal{E}^+$  as follows:

$$E_{\text{PGBA}}(\mathbf{T}, \mathbf{d}) = \sum_{(i,j) \in \mathcal{E}^*} \|\mathbf{u}_{ij}^* - \Pi(\mathbf{T}_{ij} \Pi^{-1}(\mathbf{u}_i, \mathbf{d}_i))\|_{\Sigma_{ij}}^2 + \sum_{(i,j) \in \mathcal{E}^+} \left\| \log(\tilde{\mathbf{T}}_{ij} \mathbf{T}_i \mathbf{T}_j^{-1}) \right\|_{\Sigma_{ij}^{\text{rel}}}^2,$$

where  $\tilde{\mathbf{T}}_{ij}$  is the relative pose in the pose graph, and  $\Sigma_{ij}^{\text{rel}}$  are relative-pose covariance from pairwise dense two-view correspondences as in [62]. The graph is optimized in  $\text{Sim}(3)$  to correct long-term scale drift.

### 3.2 IMU Initialization

At the beginning of each sequence, we initialize the IMU using a carefully designed three-stage procedure to ensure stable visual-inertial coupling.

**Stage 1: Pure Vision Initialization.** Using the first  $N_{\text{init}}^{\text{vis}}$  keyframes, we minimize Eq. (1) to estimate poses  $\mathbf{T}_i$  and disparities  $\mathbf{d}_i$  up to a single global scale.

**Stage 2: Inertial-Only Optimization.** We continue visual tracking until  $N_{\text{init}}^{\text{iner}}$  keyframes are available. We solve the inertial objective Eq. (2) restricted to only the  $\mathbf{R}_{\text{wg}}$ , thereby aligning the gravity direction. Keeping  $\mathbf{T}_i$  fixed, we augment the optimization variables with per-keyframe velocities  $\mathbf{v}_i$ , and IMU bias  $\mathbf{b}_i$  together with a global log-scale parameter to recover metric scale. Details are in the supplementary material.

**Stage 3: Visual-Inertial Optimization.** We further refine the estimates by jointly minimizing the visual and inertial objectives in Eqs. 1 and 2, respectively. Our staged initialization enhances robustness by postponing visual-inertial coupling until the IMU parameters can be reliably estimated.

### 3.3 Gaussian Splatting Mapping

**Preliminary.** We utilize the 3D Gaussian representation [27] to model scene appearance and geometry. The scene is represented by a set of anisotropic Gaussians  $\mathcal{G} = \{g_i\}_{i=1}^K$ . Each Gaussian  $g_i$  contains color  $\mathbf{c}_i \in \mathbb{R}^3$ , opacity  $o_i \in [0, 1]$ , mean  $\boldsymbol{\mu}_i \in \mathbb{R}^3$ , and covariance matrix  $\Sigma_i \in \mathbb{R}^{3 \times 3}$ . The color of each pixel in the rendered image is calculated by alpha-blending the visible Gaussians. Following prior work [40, 60], we replace spherical-harmonic colors with direct RGB value, reducing optimization complexity.

**Map Management.** We initialize the Gaussian map by unprojecting the new keyframe’s depth  $\mathbf{D}$  (converted from disparity  $\mathbf{d}$ ) into 3D, setting each Gaussian’s color  $\mathbf{c}$  to the corresponding pixel color and its opacity  $o$  to 0.5. For each new keyframe, we run 10 mapping iterations. In each iteration, we randomly

sample keyframes from the frontend tracking frame graph  $\mathcal{E}$  along with two global keyframes to render color  $\hat{I}$  and depth  $\hat{D}$  from the Gaussian map. We calculate color loss  $\mathcal{L}_c = \|\hat{I} - I\|_1$  and depth loss  $\mathcal{L}_d = \|\hat{D} - D\|_1$ , as well as isotropic regularization loss  $\mathcal{L}_{iso}$  [40] to prevent excessive elongation in sparsely observed regions. The optimization minimizes the following weighted total loss:

$$\mathcal{L} = \lambda_d \mathcal{L}_d + \lambda_c \mathcal{L}_c + \lambda_{iso} \mathcal{L}_{iso}. \quad (5)$$

**Loop Closure Gaussian Update.** After loop closure, the poses and scales of all keyframes in the relative pose graph are updated. To keep the map consistent without reinitializing and reoptimizing all Gaussians, we propagate each keyframe’s update to the Gaussians anchored to it. For keyframe  $k$ , let pre-/post-PGBA poses be  $(\mathbf{R}_k^-, \mathbf{p}_k^-)$  and  $(\mathbf{R}_k^+, \mathbf{p}_k^+)$ , and let the scale change be  $\delta s_k$ . For any Gaussian  $g_i$  initialized from keyframe  $k$  with mean  $\mu_i^-$  and covariance  $\Sigma_i^-$ , we update by first transforming into the old camera frame, applying the scale, then mapping to the new world frame as follows:

$$\mathbf{x}_{loc}^- = (\mathbf{R}_k^-)^\top (\mu_i^- - \mathbf{t}_k^-), \quad \mathbf{x}_{loc}^+ = \delta s_k \mathbf{x}_{loc}^-, \quad \mu_i^+ = \mathbf{R}_k^+ \mathbf{x}_{loc}^+ + \mathbf{t}_k^+. \quad (6)$$

Covariances are updated analogously:

$$\Sigma_i^+ = \mathbf{R}_k^+ \left( \delta s_k^2 (\mathbf{R}_k^-)^\top \Sigma_i^- \mathbf{R}_k^- \right) (\mathbf{R}_k^+)^\top, \quad (7)$$

while  $\mathbf{o}_i$  and  $\mathbf{c}_i$  remain unchanged. This entire update is applied in batch operations for high efficiency.

## 4 Experiments

**Datasets.** We evaluate on the EuRoC [3], RPNG AR Table [6], UTMM [53], FAST-LIVO2 [65] datasets, as well as a self-captured dataset. The EuRoC dataset provides grayscale images, while the others offer RGB ones. For FAST-LIVO2 [65] dataset, we use the poses from FAST-LIVO2 [65] as ground truth since it leverages LiDAR measurements. For our self-captured dataset, we use the Manifold Odin 1 [38], whose offline processing tool MindCloud [39] provides ground-truth poses via LiDAR-visual-inertial fusion. The remaining datasets provide motion-capture ground truth. Additional details are provided in the supp. material.

**Baselines.** We compare our VIGS-SLAM with 15 methods. (a) *Classic Visual SLAM*: DSO [11], SVO [13]; (b) *Learning-based Visual SLAM*: TartanVO [57], DROID-SLAM [54]; (c) *Visual Gaussian Splatting SLAM*: Splat-SLAM [50] and HI-SLAM2 [62]; (d) *classic Visual-Inertial SLAM*: MSCKF [43], OKVIS [30], VINS-Mono [46], OPEN-VINS [14], ORB-SLAM3 [4]; (e) *Learning-based Visual-Inertial SLAM*: DBA-Fusion [67]; (f) *Visual-Inertial Gaussian Splatting SLAM*: MM3DGS-SLAM [53], VINGS-Mono [58]; (g) *Feed-forward SLAM*: TTT3R [7].

To evaluate the online setting, for DROID-SLAM [54], Splat-SLAM [50], HI-SLAM2 [62], and our VIGS-SLAM, we report metrics computed before the final global bundle adjustment and the final color refinement (which typically takes

**Table 1: Tracking Performance on EuRoC Dataset [3]** (ATE RMSE ↓ [cm]). Best results are highlighted as **first**, **second**, and **third**. ‘F’ indicates failure. Results for SVO [13], TartanVO [57], DSO [11], MSCKF [43], OKVIS [30], VINS-Mono [46], and ORB-SLAM3 [4] are as reported by the ORB-SLAM3 paper [4]; DROID-SLAM [54] numbers are from its paper. All other results are reproduced from their official code.

| Method                  | MH_01       | MH_02       | MH_03       | MH_04       | MH_05  | V1_01  | V1_02       | V1_03  | V2_01       | V2_02       | V2_03  | Avg.        |
|-------------------------|-------------|-------------|-------------|-------------|--------|--------|-------------|--------|-------------|-------------|--------|-------------|
| <i>RGB</i>              |             |             |             |             |        |        |             |        |             |             |        |             |
| SVO [13]                | 10.00       | 12.00       | 41.00       | 43.00       | 30.00  | 7.00   | 21.00       | F      | 11.00       | 11.00       | 108.00 | N/A         |
| Splat-SLAM [50]         | 257.64      | 266.02      | 312.58      | 458.14      | 360.86 | 168.99 | 166.65      | 128.67 | 198.84      | 195.85      | 190.86 | 245.01      |
| TTT3R [7]               | 421.47      | 385.94      | 293.01      | 414.59      | 381.05 | 155.43 | 128.70      | 118.06 | 141.96      | 93.56       | 101.81 | 239.60      |
| TartanVO [57]           | 63.90       | 32.50       | 55.00       | 115.30      | 102.10 | 44.70  | 38.90       | 62.20  | 43.30       | 74.90       | 115.20 | 68.00       |
| DSO [11]                | 4.60        | 4.60        | 17.20       | 381.00      | 11.00  | 8.90   | 10.70       | 90.30  | 4.40        | 13.20       | 115.20 | 60.10       |
| DROID-SLAM [54]         | 16.30       | 12.10       | 24.20       | 39.90       | 27.00  | 10.30  | 16.50       | 15.80  | 10.20       | 11.50       | 20.40  | 18.60       |
| HI-SLAM2 [62]           | 2.66        | 1.44        | 2.71        | 6.86        | 5.07   | 3.55   | 1.32        | 2.49   | 2.56        | 1.77        | 1.92   | 2.94        |
| <i>RGB+IMU</i>          |             |             |             |             |        |        |             |        |             |             |        |             |
| MSCKF [43]              | 42.00       | 45.00       | 23.00       | 37.00       | 48.00  | 34.00  | 20.00       | 67.00  | 10.00       | 16.00       | 113.00 | 41.40       |
| VINGS-Mono [58]         | 21.03       | 16.47       | 25.46       | 25.03       | 36.01  | 6.54   | 9.79        | 11.46  | 11.51       | 93.44       | 12.39  | 24.47       |
| OKVIS [30]              | 16.00       | 22.00       | 24.00       | 34.00       | 47.00  | 9.00   | 20.00       | 24.00  | 13.00       | 16.00       | 29.00  | 23.10       |
| DBA-Fusion [67]         | 17.88       | 16.72       | 24.03       | 23.57       | 27.81  | 20.11  | 9.51        | 8.89   | 10.37       | 11.51       | 16.22  | 16.97       |
| OPEN-VINS [14]          | 50.61       | 5.61        | 7.16        | 6.34        | 12.78  | 17.46  | 9.48        | 5.96   | 11.68       | 10.47       | 14.83  | 13.85       |
| VINS-Mono [46]          | 8.40        | 10.50       | 7.40        | 12.20       | 14.70  | 4.70   | 6.60        | 18.00  | 5.60        | 9.00        | 24.40  | 11.00       |
| ORB-SLAM3 [4]           | 6.20        | 3.70        | 4.60        | 7.50        | 5.70   | 4.90   | 1.50        | 3.70   | 4.20        | 2.10        | 2.70   | 4.30        |
| <b>VIGS-SLAM (Ours)</b> | <b>1.42</b> | <b>1.29</b> | <b>2.55</b> | <b>5.16</b> | 5.64   | 3.67   | <b>1.15</b> | 2.68   | <b>2.34</b> | <b>1.53</b> | 3.27   | <b>2.79</b> |

**Table 2: Tracking Performance on RPNG AR Table Dataset [6]** (ATE RMSE ↓ [cm]). Best results are highlighted as **first**, **second**, and **third**. ‘F’ indicates failure.

| Method                  | table_01    | table_02    | table_03    | table_04    | table_05    | table_06    | table_07    | table_08    | Avg.        |
|-------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>RGB</i>              |             |             |             |             |             |             |             |             |             |
| HI-SLAM2 [62]           | 1.43        | 1.66        | 1.23        | 2.59        | F           | 1.47        | <b>0.97</b> | <b>2.67</b> | N/A         |
| Splat-SLAM [50]         | 22.46       | 41.50       | 37.93       | <b>1.09</b> | <b>1.19</b> | 1.53        | 1.76        | 4.37        | 13.98       |
| DROID-SLAM [54]         | 9.59        | 6.58        | 7.28        | 11.82       | 6.24        | 4.22        | 4.27        | 46.59       | 12.07       |
| <i>RGB+IMU</i>          |             |             |             |             |             |             |             |             |             |
| OKVIS [30]              | 9.00        | 7.70        | 15.30       | 16.20       | 24.50       | 10.20       | 13.60       | 19.80       | 14.60       |
| VINGS-Mono [58]         | 4.57        | 2.90        | 4.04        | 6.51        | <b>1.90</b> | 5.16        | 5.08        | 23.05       | 6.65        |
| DBA-Fusion [67]         | 4.73        | <b>2.77</b> | 4.85        | 6.64        | 2.27        | 4.28        | 5.64        | 20.71       | 6.49        |
| OPEN-VINS [14]          | 4.29        | 3.03        | 3.11        | 6.20        | 3.85        | 4.45        | 6.58        | 9.20        | 5.09        |
| ORB-SLAM3 [4]           | 2.52        | 15.79       | <b>1.57</b> | <b>1.22</b> | 7.34        | <b>1.49</b> | <b>1.24</b> | <b>3.43</b> | <b>4.33</b> |
| VINS-Mono [46]          | <b>2.72</b> | 5.98        | 3.30        | 4.01        | 2.18        | 1.87        | 2.05        | 5.54        | 3.46        |
| <b>VIGS-SLAM (Ours)</b> | <b>1.31</b> | <b>1.57</b> | <b>1.22</b> | 1.75        | 1.28        | <b>1.38</b> | 1.08        | 3.86        | <b>1.68</b> |

over 10 minutes). Results with these refinements are provided in the supplementary material. For brevity, we report all methods on the EuRoC dataset [3] and evaluate only the stronger and representative baselines in subsequent experiments. For DBA-Fusion [67] and its successor VINGS-Mono [58], we worked closely with the first author of VINGS-Mono [58] and made targeted modifications to improve their performance (see supplementary material for details). As confirmed by its authors, MM3DGS-SLAM [53] primarily targets an RGB+LiDAR+IMU setup; the open-sourced code does not fully support a pure visual-inertial setting. It is evaluated only on the UTMM dataset [53], and we copy the tracking results from the paper.

**Metrics.** For camera tracking, we align the estimated trajectory to the ground truth using *evo* [15], and report the Absolute Trajectory Error (ATE) in terms of RMSE [51]. In addition, we report *Recall* – percentage of ground-truth poses whose translation error to the trajectory is below a threshold. For rendering

evaluation, we report PSNR, SSIM, and LPIPS on frames that are not used as keyframes by any method, excluding all views involved in mapping. Consequently, the rendering results reported by MM3DGS-SLAM [53] are not directly comparable to ours. In the tables, we use ‘F’ to denote failure, either unable to initialize or significant drift.

#### 4.1 Mapping, Tracking, and Rendering

**EuRoC Dataset [3].** As shown in Table 1, our approach achieves the best overall ATE, ranking first or second in most sequences. In contrast, purely feedforward methods like TTT3R [7] accumulate large drift over time. ORB-SLAM3 [4], a highly engineered system with robust optimization and loop-closure mechanisms, lags behind our method. We attribute this to its reliance on handcrafted sparse features and non-differentiable components, which limit its robustness. In contrast, our system leverages dense, learning-based visual correspondences and tightly couples inertial constraints into a unified framework.



**Fig. 3: Novel View Synthesis Results across Datasets.** Sequences are sampled from RPNG [6] (table\_01, table\_06), UTM [53] (EgoDrv, Sq-2), and FAST-LIVO2 [65] (CBD2, Retail) datasets.

**Table 3: Rendering Evaluation across Datasets.** As HI-SLAM2 [62] fails on one sequence in the RPNG dataset [6], we compute averages over the remaining sequences. Detailed per-sequence results are provided in the supplementary material.

| Metric             | Method                  | RPNG         | UTMM         | FAST-LIVO2   |
|--------------------|-------------------------|--------------|--------------|--------------|
| PSNR $\uparrow$    | VINGS-Mono [58]         | 11.03        | 11.85        | 10.36        |
|                    | Splat-SLAM [50]         | 17.32        | 13.56        | 13.96        |
|                    | HI-SLAM2 [62]           | 21.12        | 18.84        | 21.49        |
|                    | <b>VIGS-SLAM (Ours)</b> | <b>22.21</b> | <b>20.87</b> | <b>23.15</b> |
| SSIM $\uparrow$    | VINGS-Mono [58]         | 0.264        | 0.408        | 0.343        |
|                    | Splat-SLAM [50]         | 0.543        | 0.470        | 0.484        |
|                    | HI-SLAM2 [62]           | 0.685        | 0.632        | 0.692        |
|                    | <b>VIGS-SLAM (Ours)</b> | <b>0.723</b> | <b>0.687</b> | <b>0.729</b> |
| LPIPS $\downarrow$ | VINGS-Mono [58]         | 0.704        | 0.660        | 0.724        |
|                    | Splat-SLAM [50]         | 0.465        | 0.653        | 0.745        |
|                    | HI-SLAM2 [62]           | 0.358        | 0.501        | 0.560        |
|                    | <b>VIGS-SLAM (Ours)</b> | <b>0.314</b> | <b>0.441</b> | <b>0.487</b> |

**Table 4: Tracking Performance on UTMM Dataset [53]** (ATE RMSE  $\downarrow$  [cm]). Best results are highlighted as **first**, **second**, and **third**. ‘F’ indicates failure. MM3DGS-SLAM [53] has not open-sourced its non-LiDAR variant; we use the paper’s reported metrics and indicate unreported sequences with ‘-’.

| Method                     | Ego-1       | Ego-2       | EgoDrv      | FastStr     | SStr-1      | SStr-2      | Sq-1        | Sq-2        | Avg.        |
|----------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>RGB</i>                 |             |             |             |             |             |             |             |             |             |
| MM3DGS-SLAM (RGB) [53]     | 4.09        | -           | 67.20       | 25.78       | -           | -           | 59.48       | -           | N/A         |
| Splat-SLAM [50]            | <b>1.38</b> | <b>0.62</b> | 3.26        | <b>0.95</b> | 5.48        | <b>0.54</b> | 103.60      | 71.29       | 23.39       |
| HI-SLAM2 [62]              | 2.06        | 3.35        | 4.36        | 0.99        | 0.71        | 0.84        | 27.85       | 24.63       | 8.10        |
| DROID-SLAM [54]            | <b>2.00</b> | 3.17        | 30.94       | 1.30        | 0.97        | 0.86        | 14.95       | <b>9.10</b> | 7.91        |
| <i>RGB+IMU</i>             |             |             |             |             |             |             |             |             |             |
| ORB-SLAM3 [4]              | 3.64        | F           | <b>3.53</b> | F           | F           | F           | F           | F           | N/A         |
| MM3DGS-SLAM (RGB+IMU) [53] | 3.41        | -           | 68.50       | 16.78       | -           | -           | 44.26       | -           | N/A         |
| VINS-Mono [46]             | 127.01      | 3.97        | 92.00       | 3.43        | <b>0.61</b> | 2.06        | 262.38      | 211.16      | 87.83       |
| DBA-Fusion [67]            | 13.27       | 5.45        | 65.01       | 118.77      | <b>0.37</b> | 1.02        | 88.93       | 133.28      | 53.26       |
| OPEN-VINS [14]             | 117.27      | 3.43        | 26.49       | 6.29        | 5.58        | 5.98        | 50.53       | 34.39       | 31.50       |
| VINGS-Mono [58]            | 5.00        | 7.29        | 12.43       | 13.36       | 0.67        | <b>0.71</b> | 36.01       | 25.70       | 12.54       |
| <b>VIGS-SLAM (Ours)</b>    | <b>1.81</b> | 0.93        | <b>1.45</b> | <b>1.20</b> | 0.81        | 0.93        | <b>2.17</b> | 16.61       | <b>3.24</b> |

**RPNG AR Table Dataset [6].** The tracking and rendering accuracies are shown in Table 2 and Table 3, respectively. The proposed VIGS-SLAM achieves the lowest tracking errors, halving the error of the second-best VINS-Mono [46], while also leading to the highest novel view synthesis scores. Qualitative results in Fig. 3 show sharper and more faithful high-frequency details (especially in the tablecloth and carpet), while baselines produce blurred textures.

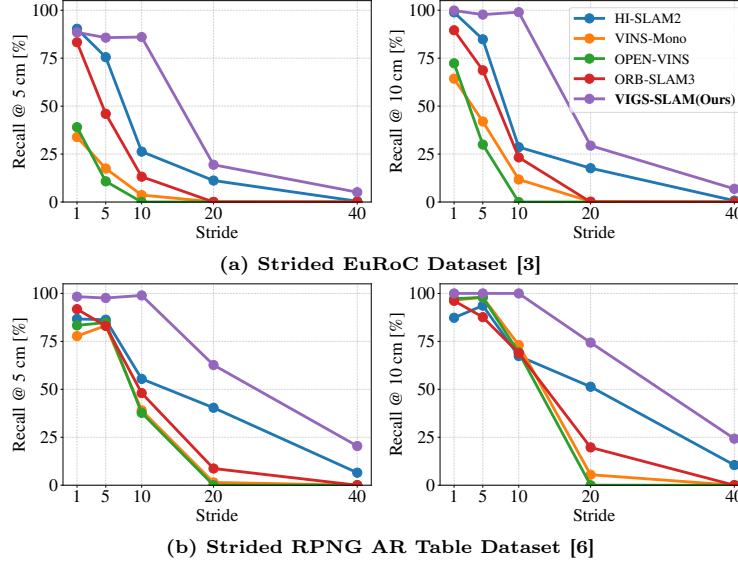
**UTMM Dataset [53].** We additionally evaluate our method on the UTMM dataset, introduced in MM3DGS-SLAM [53]. Since the non-LiDAR variant of MM3DGS-SLAM has not been publicly released, we compare against the results reported in their paper. As shown in Table 4, our method achieves substantially more accurate tracking results than baselines. Rendering results are shown in Fig. 3 and Table 3. This dataset is particularly challenging due to its cluttered backgrounds with fine-grained structures, coupled with foreground objects that

**Table 5: Tracking Performance on FAST-LIVO2 Dataset [65]** (ATE RMSE ↓ [cm]). Best results are highlighted as **first**, **second**, and **third**. ‘F’ indicates failure.

| Method                  | CBD1        | CBD2        | HKU         | Retail      | SYSU1       | Avg.        |
|-------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>RGB</i>              |             |             |             |             |             |             |
| Splat-SLAM [50]         | 5.52        | 7.74        | 4.44        | 212.01      | 313.56      | 108.65      |
| DROID-SLAM [54]         | 72.48       | 15.67       | 15.10       | 16.81       | 50.94       | 34.20       |
| HI-SLAM2 [62]           | <b>4.38</b> | 24.30       | 4.87        | <b>7.20</b> | 10.36       | 10.22       |
| <i>RGB+IMU</i>          |             |             |             |             |             |             |
| OPEN-VINS [14]          | F           | F           | F           | F           | F           | N/A         |
| ORB-SLAM3 [4]           | F           | F           | F           | F           | F           | N/A         |
| VINS-Mono [46]          | F           | F           | F           | 25.64       | F           | N/A         |
| DBA-Fusion [67]         | 22.96       | 127.79      | 56.20       | 26.69       | 337.86      | 114.30      |
| VINGS-Mono [58]         | 20.54       | 129.55      | 52.42       | 51.67       | 269.83      | 104.80      |
| <b>VIGS-SLAM (Ours)</b> | 4.50        | <b>5.76</b> | <b>3.88</b> | 8.88        | <b>7.36</b> | <b>6.08</b> |

introduce strong depth discontinuities and complex occlusions. Nonetheless, our method produces sharp renderings with minimal floating artifacts.

**FAST-LIVO2 Dataset [65].** We further evaluate on the challenging FAST-LIVO2 dataset, which features outdoor sequences with low frame rates, shaky motion, and reflective surfaces. As shown in Table 5, classical VIO systems exhibit frequent initialization failures despite repeated restarts. Our method robustly initializes from the first frame and maintains stable tracking throughout all sequences, achieving the lowest average tracking error. Rendering results in Table 3 and Fig. 3 show sharp and consistent reconstructions with preserved text and edges, benefiting from effective loop-closure handling that maintains a consistent Gaussian map, especially in outdoor sequences with large loops.



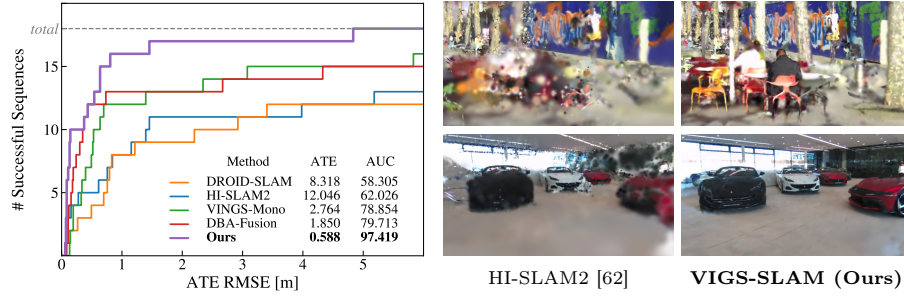
**Fig. 4: Average Tracking Performance on Strided Datasets.** We plot mean recall at 5 cm and 10 cm thresholds under different stride settings.



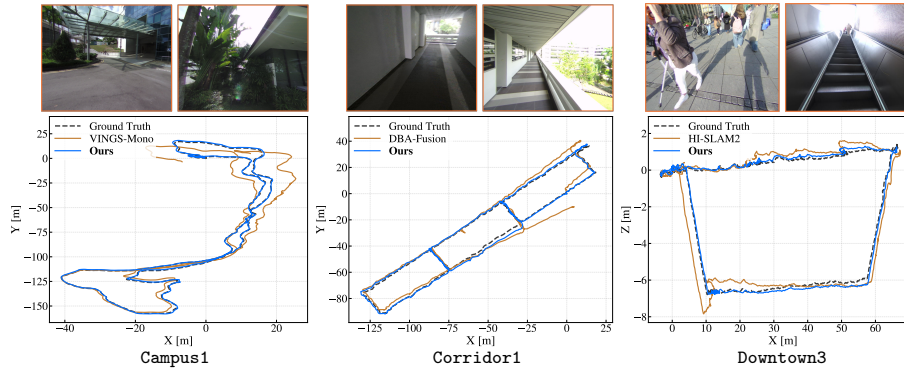
## 4.2 Tracking Robustness

**Strided Evaluation.** To evaluate robustness under degraded visual input, we create *strided* variants of the EuRoC [3] and RPNG AR Table [6] datasets by temporally subsampling RGB frames with different strides while keeping the original IMU readings, simulating frame drops, limited bandwidth and high-speed motion. We report average Recall@5cm and Recall@10cm instead of ATE RMSE, because several methods fail on certain sequences, making it difficult to compute meaningful ATE averages. We assign a recall score of zero when a method fails to produce a valid trajectory. As shown in Fig. 4, our method remains stable and consistently outperforms all baselines, whereas others frequently fail to initialize or lose tracking under large inter-frame gaps.

**Self-Captured Dataset.** We further capture 18 challenging sequences spanning (i) appearance degradation, including motion blur, exposure changes, sun glare, and low-light conditions; (ii) geometric ambiguity, such as low texture, repetitive patterns, and reflective or transparent surfaces; (iii) interference caused by dynamic objects; (iv) aggressive camera motion with hand-held shake; and (v)



**Fig. 5: Evaluation on the Self-Captured Dataset.** Left: cumulative success curve under different ATE RMSE thresholds, together with the average ATE RMSE and the AUC up to 6 meters. Right: renderings from extrapolated views.



**Fig. 6: Sample Input Frames and Trajectory Plots on the Self-Captured Dataset.** Representative frames illustrate the various challenges of our dataset.

**Table 6: Tracking Ablation on EuRoC (stride = 10) [3] Dataset.**

|                                         | Avg. ATE RMSE [cm] ↓ | Avg. Recall @ 10cm [%] ↑ |
|-----------------------------------------|----------------------|--------------------------|
| (a) w/o IMU Bias Estimation             | 338.99               | 0.05                     |
| (b) w/o IMU Fusion                      | 88.54                | 32.50                    |
| (c) w/o 3-Staged IMU Initialization     | 50.65                | 68.32                    |
| (d) w/o KF Pose Initialization with IMU | 40.94                | 64.95                    |
| (e) w/o Loop Closure                    | 23.78                | 40.63                    |
| (f) w/o Per-KF Bias Estimation          | 8.83                 | 90.01                    |
| <b>Ours (Full System)</b>               | <b>3.39</b>          | <b>98.99</b>             |

**Table 7: Rendering Ablation on FAST-LIVO2 Dataset [65].**

| Method                     | Avg. PSNR ↑  | Avg. SSIM ↑  | Avg. LPIPS ↓ |
|----------------------------|--------------|--------------|--------------|
| w/o Loop Closure GS Update | 22.06        | 0.677        | 0.541        |
| <b>Ours (Full System)</b>  | <b>23.15</b> | <b>0.729</b> | <b>0.487</b> |

long trajectories. As shown in Fig. 5 and Fig. 6, baseline methods suffer from significant tracking drift, while our method remains accurate and robust despite these severe real-world challenges. Please refer to the supplementary material for more results and dataset details.

### 4.3 Ablation Study

**Tracking.** We report ablation results for 6 design choices in Table 6. In (a) *w/o IMU Bias Estimation*, the IMU bias is fixed to zero and not optimized. In (b) *w/o IMU Fusion*, the optimization is constrained solely by visual residuals. In (c), we remove the inertial-only optimization stage during IMU initialization. In (d) *w/o KF Pose Initialization with IMU*, we disable the IMU pre-integration-based pose initialization for newly selected keyframes. In (e), we disable the loop closure. In (f), we optimize a single global IMU bias shared across all keyframes instead of estimating per-keyframe biases. Table 6 demonstrates that removing any component will degrade tracking accuracy as well as robustness, while our full system achieves the best results.

**Mapping.** We ablate the loop closure Gaussian update in Table 7. Disabling this update leads to inconsistent Gaussian maps, which in turn leads to rendering performance degradation.

## 5 Conclusion

We present VIGS-SLAM, a novel visual-inertial 3D Gaussian Splatting SLAM system that achieves robust real-time tracking and high-fidelity Gaussian mapping. It tightly couples learning-based dense visual correspondences with inertial constraints within a unified optimization framework. Furthermore, we perform robust IMU initialization and efficient loop closure with consistent Gaussian updates. Extensive evaluations on five challenging datasets show that VIGS-SLAM achieves state-of-the-art performance in both tracking accuracy and novel view

synthesis quality, and is among the very few methods that succeed on all sequences without failure. Code and dataset will be made public.

**Limitation.** In the current system, the Gaussian map does not directly improve tracking. Incorporating a Gaussian re-rendering loss for pose optimization yields only marginal gains, likely because renderings during online optimization are not yet sufficiently sharp to produce reliable gradients for pose refinement.

## 6 Acknowledgements

We thank Ke Wu for help with VINGS-Mono and Codey Sun for help with MM3DGS-SLAM. We are grateful to Anusha Krishnan and Shaohui Liu for valuable discussions, and to Boyang Sun and Zuria Bauer for providing the capture site for the demo.

## References

1. Belos, T., Monasse, P., Dokladalova, E.: Mod slam: Mixed method for a more robust slam without loop closing. In: VISAPP (2022)
2. Bloesch, M., Omari, S., Hutter, M., Siegwart, R.: Robust visual inertial odometry using a direct ekf-based approach. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 298–304. IEEE (2015)
3. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R.: The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research* **35**(10), 1157–1163 (2016)
4. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE transactions on robotics* **37**(6), 1874–1890 (2021)
5. Cao, Z., Wu, H., Tang, L.W., Luo, Z., Zhu, Z., Zhang, W., Pollefeys, M., Oswald, M.R.: Mcgs-slam: A multi-camera slam framework using gaussian splatting for high-fidelity mapping. *arXiv preprint arXiv:2509.14191* (2025)
6. Chen, C., Geneva, P., Peng, Y., Lee, W., Huang, G.: Monocular visual-inertial odometry with planar regularities. In: *Proc. of the IEEE International Conference on Robotics and Automation*. London, UK (2023)
7. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Ttt3r: 3d reconstruction as test-time training. In: *The Fourteenth International Conference on Learning Representations* (2026)
8. Chung, C.M., Tseng, Y.C., Hsu, Y.C., Shi, X.Q., Hua, Y.H., Yeh, J.F., Chen, W.C., Chen, Y.T., Hsu, W.H.: Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping. *arXiv preprint arXiv:2209.13274* (2022)
9. Dellaert, F.: Factor graphs and gtsam: A hands-on introduction. *Georgia Institute of Technology, Tech. Rep* **2**(4) (2012)
10. Deng, K., Zhang, Y., Yang, J., Xie, J.: Gigaslam: Large-scale monocular slam with hierarchical gaussian splats. *arXiv preprint arXiv:2503.08071* (2025)
11. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* (2017)
12. Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D.: Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In: *RSS* (2015)

13. Forster, C., Pizzoli, M., Scaramuzza, D.: Svo: Fast semi-direct monocular visual odometry. In: 2014 IEEE international conference on robotics and automation (ICRA). pp. 15–22. IEEE (2014)
14. Geneva, P., Eckenhoff, K., Lee, W., Yang, Y., Huang, G.: Openvins: A research platform for visual-inertial estimation. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 4666–4672. IEEE (2020)
15. Grupp, M.: evo: Python package for the evaluation of odometry and slam. <https://github.com/MichaelGrupp/evo> (2017)
16. Ha, S., Yeon, J., Yu, H.: Rgb-d gs-icp slam. arXiv preprint arXiv:2403.12550 (2024)
17. Hu, J., Chen, X., Feng, B., Li, G., Yang, L., Bao, H., Zhang, G., Cui, Z.: Cg-slam: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field. In: Proc. of the European Conf. on Computer Vision (ECCV) (2024)
18. Huang, G.P., Mourikis, A.I., Roulletiotis, S.I.: Observability-based rules for designing consistent ekf slam estimators. *The international journal of Robotics Research* **29**(5), 502–528 (2010)
19. 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. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
20. Inc., A.: Apple vision pro (2024), <https://www.apple.com/apple-vision-pro/specs>
21. Inc., D.T.: Dji osmo action cameras (gyroflow support) (2024), <https://docs.gyroflow.xyz/app/getting-started/supported-cameras/dji>
22. Inc., I.: Insta360 x3 action camera (2022), <https://insta360.com/product/insta360-x3>
23. Inc., S.: Zed 2 stereo camera (2024), <https://www.stereolabs.com/products/zed-2>
24. Johari, M.M., Carta, C., Fleuret, F.: Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. arXiv preprint arXiv:2211.11704 (2022)
25. 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. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 10230–10237. IEEE (2024)
26. 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. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
27. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. on Graphics* (2023)
28. Kruzhkov, E., Savinykh, A., Karpyshev, P., Kurenkov, M., Yudin, E., Potapov, A., Tsetserukou, D.: Meslam: Memory efficient slam based on neural fields. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC) (2022)
29. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 2548–2555 (2011)
30. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research* **34**(3), 314–334 (2015)
31. Li, K., Niemeyer, M., Navab, N., Tombari, F.: Dns slam: Dense neural semantic-informed slam. arXiv preprint arXiv:2312.00204 (2023)
32. Li, L., Zhang, L., Wang, Z., Shen, Y.: Gs3lam: Gaussian semantic splatting slam. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 3019–3027 (2024)

33. Li, M., Liu, S., Zhou, H., Zhu, G., Cheng, N., Deng, T., Wang, H.: Sgs-slam: Semantic gaussian splatting for neural dense slam. In: Proc. of the European Conf. on Computer Vision (ECCV) (2024)
34. Li, M., Mourikis, A.I.: High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research* **32**(6), 690–711 (2013)
35. Liu, H., Chen, M., Zhang, G., Bao, H., Bao, Y.: Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1974–1982 (2018)
36. Liu, X., Tan, N.: Gi-slam: Gaussian-inertial slam. arXiv preprint arXiv:2503.18275 (2025)
37. LLC, G.: Google pixel 9 (2024), [https://store.google.com/product/pixel\\_9\\_specs?hl=en-US](https://store.google.com/product/pixel_9_specs?hl=en-US)
38. Manifold Tech: Manifold Odin 1: Spatial Memory Module. <https://www.manifoldtech.cn/products/Odin1>
39. Manifold Tech: Mindcloud: A point cloud data processing platform. <https://www.manifold.com.co/mindcloud>
40. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
41. Meta Platforms, I.: Meta quest 3 (2023), <https://vr-compare.com/headset/metaquest3>
42. 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: Proc. of the European Conf. on Computer Vision (ECCV) (2020)
43. Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint kalman filter for vision-aided inertial navigation. In: Proceedings 2007 IEEE international conference on robotics and automation. pp. 3565–3572. IEEE (2007)
44. Ndoye, A., Nègre, A., Marchand, N., Ruffier, F.: Vigs-fusion: Fast gaussian splatting slam processed onboard a small quadrirotor. In: International Conference on Advanced Robotics (ICAR) (2025)
45. Peng, Z., Shao, T., Liu, Y., Zhou, J., Yang, Y., Wang, J., Zhou, K.: Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In: SIGGRAPH 2024 Conference Papers (2024)
46. Qin, T., Li, P., Shen, S.: Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE transactions on robotics* **34**(4), 1004–1020 (2018)
47. 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)
48. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 2564–2571 (2011)
49. Samsung Electronics Co., L.: Samsung galaxy s24 (2024), [https://www.samsung.com/latin\\_en/smartphones/galaxy-s24/specs](https://www.samsung.com/latin_en/smartphones/galaxy-s24/specs)
50. Sandström, E., Tateno, K., Oechsle, M., Niemeyer, M., Van Gool, L., Oswald, M.R., Tombari, F.: Splat-slam: Globally optimized rgb-only slam with 3d gaussians. arXiv preprint arXiv:2405.16544 (2024)
51. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS) (2012)
52. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: imap: Implicit mapping and positioning in real-time. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2021)

53. Sun, L.C., Bhatt, N.P., Liu, J.C., Fan, Z., Wang, Z., Humphreys, T.E., Topcu, U.: Mm3dgs slam: Multi-modal 3d gaussian splatting for slam using vision, depth, and inertial measurements (2024)
54. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2021)
55. Von Stumberg, L., Usenko, V., Cremers, D.: Direct sparse visual-inertial odometry using dynamic marginalization. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 2510–2517. IEEE (2018)
56. Wang, H., Wang, J., Agapito, L.: Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2023)
57. Wang, W., Hu, Y., Scherer, S.: Tartanvo: A generalizable learning-based vo. In: *Conference on Robot Learning*. pp. 1761–1772. PMLR (2021)
58. Wu, K., Zhang, Z., Tie, M., Ai, Z., Gan, Z., Ding, W.: Vings-mono: Visual-inertial gaussian splatting monocular slam in large scenes. *IEEE Transactions on Robotics* (2025)
59. Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2024)
60. Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070* (2023)
61. Zhai, H., Huang, G., Hu, Q., Li, G., Bao, H., Zhang, G.: Nis-slam: Neural implicit semantic rgb-d slam for 3d consistent scene understanding. *IEEE Transactions on Visualization and Computer Graphics* (2024)
62. Zhang, W., Cheng, Q., Skuddis, D., Zeller, N., Cremers, D., Haala, N.: Hi-slam2: Geometry-aware gaussian slam for fast monocular scene reconstruction. *IEEE Transactions on Robotics* **41**, 6478–6493 (2025)
63. Zhang, W., Sun, T., Wang, S., Cheng, Q., Haala, N.: Hi-slam: Monocular real-time dense mapping with hybrid implicit fields. *IEEE Robotics and Automation Letters* (2023)
64. Zhang, Y., Wang, D., Xu, J., Liu, M., Zhu, P., Ren, W.: Nerf-vio: Map-based visual-inertial odometry with initialization leveraging neural radiance fields. *arXiv preprint arXiv:2503.07952* (2025)
65. Zheng, C., Xu, W., Zou, Z., Hua, T., Yuan, C., He, D., Zhou, B., Liu, Z., Lin, J., Zhu, F., et al.: Fast-livo2: Fast, direct lidar-inertial-visual odometry. *IEEE Transactions on Robotics* (2024)
66. Zheng, J., Zhu, Z., Bieri, V., Pollefeys, M., Peng, S., Armeni, I.: Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 11461–11471 (2025)
67. Zhou, Y., Li, X., Li, S., Wang, X., Feng, S., Tan, Y.: Dba-fusion: Tightly integrating deep dense visual bundle adjustment with multiple sensors for large-scale localization and mapping. *IEEE Robotics and Automation Letters* **9**(7), 6138–6145 (2024)
68. Zhu, L., Li, Y., Sandström, E., Schindler, K., Armeni, I.: Loopsplat: Loop closure by registering 3d gaussian splats. In: *Proc. of the International Conf. on 3D Vision (3DV)* (2025)
69. Zhu, S., Wang, G., Blum, H., Liu, J., Song, L., Pollefeys, M., Wang, H.: Sni-slam: Semantic neural implicit slam. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2024)

70. Zhu, Z., Peng, S., Larsson, V., Cui, Z., Oswald, M.R., Geiger, A., Pollefeys, M.: Nicer-slam: Neural implicit scene encoding for rgb slam. In: Proc. of the International Conf. on 3D Vision (3DV) (2024)
71. 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: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2022)