International Journal of Innovative Computing, Information and Control
Volume **22**, Number **3**, June **2026**

ICIC International ©2026 ISSN 1349-4198
pp. **639–652**

MTCGS: MULTI-MODAL TIGHTLY COUPLED INERTIAL MEASUREMENT UNIT 3D GAUSSIAN SPLATTING SLAM

Fenghua Wang[1,2,*], Yachao Hu[1,2], Zhicheng Xu[1,2] and Yixin Zhang[1]

[1]Qingdao Institute of Software
College of Computer Science and Technology
[2]The Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software
China University of Petroleum (East China)
No. 66, Changjiang West Road, Huangdao District, Qingdao 266580, P. R. China
{ z23070168; Z22070096; 2207020525 }@s.upc.edu.cn
*Corresponding author: fenghuawang@upc.edu.cn

Received July 2025; revised November 2025

Abstract. *Simultaneous Localization and Mapping (SLAM) based on 3D Gaussian Splatting (3DGS) have demonstrated significant advantages in high-fidelity map reconstruction and real-time rendering. In particular, Inertial Measurement Unit (IMU) has proven to be highly effective in enhancing SLAM performance by providing robust motion constraints in visually degraded environments. However, existing visual-inertial SLAM systems using 3DGS often suffer from poor initialization and insufficient IMU bias correction, which limits their robustness in challenging scenarios. To address these issues, we propose MTCGS, a tightly coupled multi-modal SLAM framework that integrates 3DGS with inertial measurements. Our method employs joint optimization of IMU pre-integration residuals and 3DGS rendering loss to refine both camera poses and inertial noise parameters, enhancing pose estimation especially in the presence of visual degradation. For map reconstruction, we introduce a hybrid keyframe selection strategy combining a sliding window with adaptive optimization to prevent catastrophic forgetting. Furthermore, 3D Gaussian smoothing and an improved isotropic regularization loss are applied to suppressing high-frequency artifacts and enhancing reconstruction quality. We evaluate our system on the OpenLORIS and TUM-RGBD datasets. Experimental results show that MTCGS outperforms the representative method MM3DGS in both localization accuracy and mapping robustness, particularly under conditions of noisy inertial data.*
Keywords: SLAM, IMU, 3DGS, Tightly coupled optimization, Multi-modal fusion

1. **Introduction.** SLAM is an important research topic in the field of computer vision, which requires exploring the environment and estimating the camera's pose in real time without prior map information. Traditional SLAM systems are suitable for large-scale scenarios such as autonomous driving, robotics, and augmented reality through high positioning accuracy and real-time processing capabilities. These advancements have significantly advanced the autonomy and interactivity of intelligent systems.

Although SLAM technology has achieved remarkable results, the discrete surface representations (e.g., point clouds [1], voxel meshes [2], and octree trees [3]) used in traditional SLAM technologies, resulting in inevitable distortions during map reconstruction. In applications requiring high-fidelity 3D reconstruction, the Neural Radiance Field (NeRF) [4] and 3DGS [5] can be used to establish a more detailed geometric field model. Compared with the traditional map representation, the quality of radiance field mapping is more realistic and it provides new perspective synthesis capability.

DOI: 10.24507/ijicic.22.03.639

Recently, 3D Gaussian Splatting has emerged as a more efficient alternative to NeRF. Compared to NeRF [6-8], 3DGS leverages explicit scene representation and rasterization-based rendering to achieve superior image quality and significantly faster rendering speed, making it well-suited for real-time SLAM applications [9-11]. Inertial measurement units, which provide visual-independent motion information, are lightweight and cost-effective, making them valuable for enhancing SLAM robustness in challenging environments. Combining the complementary strengths of 3DGS and IMUs, visual-inertial SLAM systems based on 3DGS [12-15] show strong potential for real-world deployment. This combination offers both high-fidelity reconstruction and real-time performance, drawing increasing attention in robotics and autonomous systems.

However, existing 3DGS-based visual-inertial SLAM systems face two challenges. First, despite the advantages of IMU integration, many systems suffer from poor initialization and unoptimized IMU bias correction, making them highly sensitive to inertial noise and calibration errors [14]. Second, in the map reconstruction process, 3DGS tends to suffer from catastrophic forgetting of old observations and the emergence of high-frequency artifacts [16,17], which degrade reconstruction quality over time.

To address these limitations, we propose MTCGS, a tightly coupled multi-modal SLAM framework that integrates 3DGS with inertial measurements. Our method performs joint optimization of IMU pre-integration residuals and 3DGS rendering loss, enabling simultaneous refinement of camera poses and IMU noise parameters. For mapping, we design a hybrid keyframe selection strategy that combines a sliding window and an adaptive optimization window to mitigate long-tail optimization issues and prevent forgetting. Additionally, to further enhance reconstruction quality, we introduce a low-pass 3D Gaussian smoothing filter with a dynamically adjusted kernel size, along with an improved isotropic regularization loss to suppress high-frequency artifacts.

Specifically, the main contributions of this paper are as follows.

- We propose a robust framework that integrates IMU data with 3D Gaussian Splatting. This includes a joint optimization algorithm combining IMU residuals and 3D Gaussian rendering loss, along with a gravity-aligned initialization method.
- We design a keyframe selection strategy that combines a sliding window mechanism with adaptive optimization to mitigate catastrophic forgetting. In addition, we introduce 3D Gaussian smoothing and an improved isotropic regularization loss to suppress high-frequency artifacts in map reconstruction.
- We implement an RGB-D SLAM system based on 3DGS, which demonstrates improved robustness to IMU data quality compared to existing methods such as MM-3DGS.

The remainder of this paper is organized as follows. Section 2 provides a review of related work in visual-inertial SLAM and 3D Gaussian Splatting. Section 3 presents the proposed MTCGS framework in detail, including the tightly coupled optimization strategy, initialization method, and the designed modules for mapping and regularization. Section 4 reports extensive experimental evaluations, where the proposed system is compared against several state-of-the-art baselines to demonstrate its effectiveness and robustness. Finally, Section 5 concludes the paper and discusses limitations as well as potential future research directions.

2. **Related Works.**

2.1. **Dense visual SLAM and map representation.**

Sparse SLAM algorithms, such as ORB-SLAM2 [18] and DSO [19], prioritize localization accuracy and computational

efficiency, typically employing sparse feature point clouds for map representation. In contrast, dense (or intensive) SLAM [20] aims for high-quality scene reconstruction, with map representations generally categorized into two paradigms: view-centric [17] and world-centric [21]. In dense view-centric SLAM, the 3D scene representation is often anchored to keyframes [22,23]. This allows for localized updates, which helps reduce computational overhead and facilitates efficient optimization. On the other hand, world-centric dense SLAM systems anchor the 3D map to a global coordinate frame, and commonly use representations such as surfaces [24], voxel grids [3], and Signed Distance Functions (SDFs) [25] to model the scene.

2.2. **SLAM based on NeRF and 3D Gaussians.** In the field of state-of-the-art view synthesis, NeRF's implicit representation method [4,26] has achieved remarkable success. iMAP [6] enables MLP based map construction for the first time, by reducing the number of parameters in NeRF and using RGB-D data. NICE-SLAM [7] adopts the explicit radiance field of multi-level network structure to optimize the parameters of scene representation hierarchically. ESLAM [8] uses three coarse feature planes and three fine feature planes to improve convergence speed and reconstruction quality.

Compared to NeRF, 3DGS [5] and subsequent developments [27] have significant advantages in terms of visual quality and training speed. The Gaussian Splatting SLAM (MonoGS) [28] adopted an explicit mapping method based on 3DGS. The analytical Jacobian matrix between pose and Gaussian features is constructed using the chain derivation rule, and the 3DGS-based SLAM system is realized. GS-SLAM [11] further introduces a dynamic scaling strategy that adaptively adds or removes Gaussian primitives based on the complexity of the scene, and achieves joint optimization of camera pose and scene geometry through global Beam Adjustment (BA).

2.3. **Visual inertial SLAM.** Pure visual SLAM systems are vulnerable to challenges such as rapid motion and overexposure. A widely adopted solution is to fuse visual data with inertial measurements from IMUs, an approach that has also been extended to radiance-field-based SLAM systems [12,13]. However, the quality of inertial data is often degraded by sensor bias and limited sampling rates. To mitigate these issues, techniques such as filtering [14] and nonlinear optimization [15,22] are commonly used. Among them, manifold-based pre-integration [29] is a popular method for preprocessing IMU data to accelerate subsequent optimization. OKVIS [30] introduces a keyframe-based sliding window approach for batch nonlinear optimization. Similarly, VINS-Mono [15] employs a tightly-coupled sliding window pose estimation framework and demonstrates that nonlinear optimization generally yields higher accuracy than filtering methods. It also proposes a complete and lightweight SLAM front-end based on optical flow.

3. **Method.** Our system framework is illustrated in Figure 1. As shown in the figure, the system takes RGB, RGB-D, and IMU data as input. In the localization module, IMU pre-integration is used to initialize motion parameters and estimate the gravity direction, and the corresponding residuals are incorporated into pose optimization. In parallel, the mapping module adopts a hybrid strategy that combines adaptive keyframe selection with a fixed sliding window to achieve a balance between real-time performance and global consistency. Moreover, a 3D Gaussian low-pass filter with an adaptive kernel size is employed to smooth Gaussian primitives, thereby suppressing high-frequency artifacts and enhancing reconstruction quality. Finally, the optimization process jointly minimizes the IMU pre-integration residuals and the 3D Gaussian rendering loss, which allows the simultaneous refinement of camera pose, IMU bias, and Gaussian parameters.

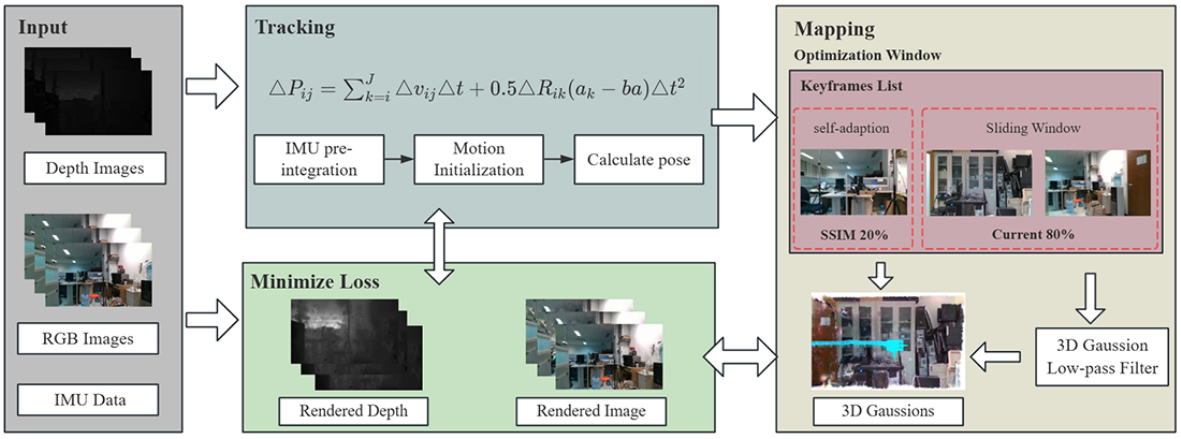

FIGURE 1. Overview of the MTCGS-SLAM framework

3.1. **3D Gaussian Splatting.** Scene mapping uses a set of 3D Gaussian features as the scene representation. Each 3D Gaussian is defined with a covariance matrix, which is parameterized as follows:

$$G = \{G_i : (\mu_i, \Sigma_i, O_i, c_i) | i = 1, 2, \ldots, N\} \tag{1}$$

The 3D Gaussian distribution can be expressed as

$$G(x) = \exp\left(-\frac{1}{2}(x-\mu)\Sigma(x-\mu)^T\right) \tag{2}$$

Color and opacity are directly represented through explicit 3D Gaussian expressions. Here $\mu_i \in R^3$ represents position, $\Sigma_i \in R^{3\times3}$ represents 3D covariance matrix, $O_i \in R$ represents opacity, and $c_i \in R^{12}$ represents spherical harmonic coefficient, which can realize color calculation. In order to reduce the number of 3DGS parameters, the covariance parameters are expressed as

$$\Sigma = RSS^TR^T \tag{3}$$

where $S \in R^3$ is a 3D scale vector, and $R \in R^{3\times3}$ is the rotation matrix, storing as a 4D quaternion. Given the camera pose $= \{R, P\}$ to generate a 2D plane for a 3D Gaussian projection:

$$\Sigma' = Jp^{-1}\Sigma\left(p^{-1}\right)^T J^T \tag{4}$$

where J is the Jacobian determinant of the affine approximation of the projective function, p denotes the projection function that maps 3D coordinates into the image plane, and Σ' is the projected 2D covariance matrix. After projection, the Gaussian features are sorted by depth and the map is rendered in an integral manner similar to Equations (5) and (6). Therefore, there are commonalities in the pose estimation algorithms of radiated field SLAM.

$$C(G,T) = \sum_{i=1}^{n} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{5}$$

where c_i and α_i are obtained by two-dimensional Gaussian projection sampling, with α_i denoting the opacity (or transparency weight) of the i-th Gaussian.

Similarly, the pixel depth D can be expressed as

$$D(G,T) = \sum_{i=1}^{n} d_i a_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{6}$$

Here d_i denotes the depth value of the i-th Gaussian sample, and a_i represents its blending weight after projection.

Since the rendering process is differentiable, 3D Gauss parameters and camera pose can be optimized by minimizing pixel depth and color loss:

$$loss_{gaussian} = L_1(I, C(G,T)) + L_1(D, D(G,T)) \tag{7}$$

where I represents the truth value of the RGB image, D represents the truth value of the depth map, and $L_1(\cdot)$ denotes the L1 loss function.

3.2. **IMU pre-integration algorithm.** Accurate pose optimization during localization requires good initial estimates for fast convergence. Traditional constant-velocity models often fail under rapid motion or low frame rates.

To improve initialization, this study integrates IMU acceleration and angular velocity data. However, direct IMU integration suffers from noise, bias, and attitude errors, which amplify over time due to gravity coupling, leading to instability.

To address this, we adopt a manifold-based IMU pre-integration algorithm [25], which jointly optimizes camera pose and IMU noise parameters by fusing IMU data with the differentiable rendering loss from 3D Gaussian Splatting.

The camera state at time T_i is defined as

$$C_i = [R_i, P_i, V_i, ba_i, bg_i] \tag{8}$$

where $R_i \in \mathbb{R}^{3\times3}$ (orientation), $P_i \in \mathbb{R}^3$ (position), $V_i \in \mathbb{R}^3$ (velocity) are motion parameters, and $ba_i, bg_i \in \mathbb{R}^3$ are the accelerometer and gyroscope biases.

IMU pre-integration computes the relative velocity, position, and orientation between frames, as well as their Jacobians with respect to ba_i and bg_i. These pre-integrated measurements and Jacobians are used in the subsequent optimization.

The specific equations for the pre-integrated terms and their Jacobians are as follows:

$$\Delta R_{ij} = \prod_{k=i}^{j} \exp((\omega_k - bg_k)\Delta t) \tag{9}$$

$$\Delta v_{ij} = \sum_{k=i}^{J} \Delta R_{ik}(a_k - ba)\Delta t \tag{10}$$

$$\Delta P_{ij} = \sum_{k=i}^{J} \Delta v_{ik}\Delta t + 0.5\Delta R_{ik}(a_k - ba)\Delta t^2 \tag{11}$$

Here, Δt denotes the time interval between two consecutive frames. ω_k represents the gyroscope's angular velocity measurement at the k-th frame, and a_k denotes the accelerometer's measurement at the k-th IMU frame. It is assumed that the bias remains constant between image frames. Here, ΔR_{ij} is the pre-integrated relative rotation from frame i to j, Δv_{ij} is the pre-integrated relative velocity, and ΔP_{ij} is the pre-integrated relative position.

To reduce the computational cost of frequent updates, the Jacobian matrix of the pre-integration terms is computed iteratively and stored, based on the assumption of zero bias during integration.

When external noise is introduced into the integration process, the covariance matrix of the pre-integrated measurements becomes a critical metric for evaluating the confidence of the pre-integration results. The initial covariance matrix is defined as a zero matrix of size 9×9. Let C_{j-1} represent the covariance matrix at the $(j-1)$-th frame. Upon

receiving the j-th IMU frame, the covariance matrix is updated recursively according to the following formula:

$$A_{j-1} = \begin{bmatrix} \Delta R_{j,j-1} & 0 & 0 \\ -\Delta R_{j,j-1}(a-ba)^{\Lambda}\Delta t & I & 0 \\ -0.5\Delta R_{j,j-1}(a-ba)^{\Lambda}\Delta t & -I\Delta t & I \end{bmatrix} \tag{12}$$

$$B_{j-1} = \begin{bmatrix} J_r^{j-1}\Delta t & 0 \\ 0 & \Delta R_{j,j-1}\Delta t \\ 0 & 0.5\Delta R_{j,j-1}\Delta t^2 \end{bmatrix} \tag{13}$$

$$C_j = A_{j-1}C_{j-1}A_{j-1}^T + B_{j-1}\Sigma_\eta B_{j-1}^T \tag{14}$$

where ΔR is pre-integration term, and J_r^{j-1} is the right Jacobian of ΔR on the $SO(3)$. $\Sigma_\eta \in R^{6\times 6}$ is the zero-mean Gaussian noise covariance matrix for acceleration and angular velocity. Here, I denotes the 3×3 identity matrix, and $(a-ba)^{\Lambda}$ represents the skew-symmetric matrix (cross-product operator) of the vector $(a-ba)$.

3.3. **Tightly coupled IMU pre-integration.** Motion data include gravity and velocity. The direction of gravity and the initial velocity have a crucial influence on the convergence result of the motion constraints loss. This paper adopts the initialization method of Vins-Mono. Before the initialization process, the IMU data is only used to perform IMU pre-integration without participating in the tracking module. Only when the accumulation of keyframes exceeds 10 frames and the image frames exceed 200 frames, the pose derived from the tracking module is used as the measurements to initialize the velocity and gravity acceleration estimates for each image frame.

In the pose optimization stage, the residual between the predicted value (provided by the motion parameter item of the previous frame) and the measured value (pre product item) is used as the motion constraint loss, and the motion parameters of the current frame and the previous frame are optimized by minimizing this loss.

The residual formula is as follows:

$$r_R = Log\left[\Delta R_{ij} Exp\left(\frac{\partial \Delta R_{ij}}{\partial bg}\Delta bg\right) R_i^T R_j\right] \tag{15}$$

$$r_v = R_i^T(v_j - v_i - g\Delta t) - \left[\Delta v_{ij} + \frac{\partial \Delta V_{ij}}{\partial bg}\Delta bg + \frac{\partial \Delta V_{ij}}{\partial ba}\Delta ba\right] \tag{16}$$

$$r_P = R_i^T\left(P_j - P_i - v_i\Delta t - 0.5g\Delta t^2\right) - \left[\Delta P_{ij} + \frac{\partial \Delta P_{ij}}{\partial bg}\Delta bg + \frac{\partial \Delta P_{ij}}{\partial ba}\Delta ba\right] \tag{17}$$

Here, $R_i \in SO(3)$ and $R_j \in SO(3)$ denote the orientations at frames i and j, respectively. $v_i, v_j \in \mathbb{R}^3$ represent the linear velocities, and $P_i, P_j \in \mathbb{R}^3$ represent the positions of frames i and j. $g \in \mathbb{R}^3$ is the gravity vector. $\Delta bg, \Delta ba \in \mathbb{R}^3$ are the gyroscope and accelerometer bias increments, respectively.

The motion constraints loss is obtained as follows:

$$loss_{IMU} = [r_R, r_v, r_P]^T \Sigma C^{-1} [r_R, r_v, r_P] \tag{18}$$

Here, ΣC^{-1} is the inverse covariance matrix of the IMU pre-integration noise, used to normalize the residuals.

Finally, this motion constraints loss can be directly applied to the tracking module of SLAM:

$$loss_{tracking} = loss_{gaussian} + \lambda loss_{IMU} \tag{19}$$

where λ is the total weight of the motion constraint loss.

To ensure that the influence of inertial measurements is adaptively modulated based on the geometric reliability of the scene, we introduce a confidence-aware weighting strategy for λ. Specifically, we define a confidence score $Con \in [0, 1]$ for each frame based on the visibility and optimization history of 3D Gaussians within the current view:

$$Con = \frac{K_{opt} \cap K_{all}}{K_{all}} \tag{20}$$

Here, K_{all} denotes the set of 3D Gaussians visible in the current frame, and K_{opt} represents the subset of those Gaussians that were actively optimized in recent mapping iterations. A higher value of Con indicates a greater level of geometric confidence in the current frame.

Based on this confidence measure, we define λ as

$$\lambda_{IMU} = 0.03 + 0.07\sqrt{1 - Con} \tag{21}$$

This adaptive weighting mechanism enables the SLAM system to better handle scenes with significant viewpoint changes or sparse geometric structures by dynamically balancing the contribution of visual and inertial modalities during optimization.

In order to accelerate the convergence of the loss, a constant velocity assumption is made on the velocity and attitude of the current frame based on the motion parameters of the previous frame, pre-product items, and gravity as a prior for pose and velocity:

$$v_j = v_i + g\Delta t + R_i \Delta v \tag{22}$$

$$P_j = P_i + 0.5g\Delta t^2 + v_i \Delta t + R_i \Delta p \tag{23}$$

Here, Δv and Δp denote the pre-integrated velocity and position increments between frames i and j, respectively.

Every time the zero bias is updated, in order to avoid recomputing the pre-integration, the Jacobian matrix of the zero bias is linearly updated through pre-integration increments, and the pre-integration sub-items are update.

3.4. **Keyframe strategy.** To ensure efficient map construction without sacrificing quality, we design a hybrid optimization strategy that combines a sliding window approach with an adaptive keyframe selection mechanism.

Let the total number of optimization iterations for each mapping round be denoted as T. We divide this budget into two components: T_s, which accounts for $\frac{4}{5}T$, is allocated to keyframes within the current sliding window, while the remaining $T_a = \frac{1}{5}T$ is reserved for adaptive keyframes outside the window. This design ensures the majority of computational resources are dedicated to temporally local and structurally relevant frames, while still allocating a portion to historically significant but low-quality keyframes.

During each mapping round, we maintain a sliding window of size W centered around the current frame. Outside this window, we evaluate the Structural Similarity Index Measure (SSIM) of all keyframes with respect to the current view. Let S_i represent the SSIM score of keyframe i. We sort all keyframes not in the sliding window by S_i in ascending order and select the top K keyframes to form an adaptive set. These are the frames most likely to be under-optimized and thus benefit from additional refinement.

To avoid repeated computation on consistently low-quality frames, we define a masking rule. For each adaptive keyframe, we track how many consecutive rounds it has been selected. If a keyframe k_i has been selected m times consecutively, we set a mask flag:

$$\text{If } C_i \geq m,\ mask(k_i) = 1 \tag{24}$$

This mechanism ensures that computational effort is not wasted on frames that consistently fail to improve.

3.5. **Low-pass filter.** In order to mitigate the influence of aliasing artifacts generated by high frequency 3D Gaussian on the quality of the construction, it is a simple and effective method to use low-pass filter to limit the maximum frequency of 3D Gaussian [28]. In this paper, we calculate the maximum sampling frequency of all 3D Gaussians every k iterations during the mapping phase. Instead of directly altering the scale of the 3D Gaussians, the low-pass filter is treated as an inherent property of each Gaussian primitive. During rendering, the filter is applied to constraining the effective size of the Gaussian, thereby limiting its frequency response. Because both the sliding window and the adaptive window are involved in the mapping phase, we compute the maximum sampling frequency using the keyframes selected by both strategies. The goal is to ensure that each 3D Gaussian primitive is reconstructed by at least one keyframe.

Recent work such as Mip-Splatting [27] mitigates aliasing by introducing multi-scale Gaussian representations. Although effective, it depends on discretized mipmap levels and view-dependent sampling, which may cause redundancy and inefficiency in complex scenes. In contrast, our method integrates the low-pass filter as an inherent property of each Gaussian and computes a global maximum sampling frequency across selected keyframes. This avoids explicit multi-scale duplication, ensuring stable reconstruction with higher efficiency and lower overhead.

We denote the maximum sampling frequency of a 3D Gaussian as

$$v = \max \left\{ \frac{f}{d_n} \right\}_{n=1}^{N} \tag{25}$$

where N represents the number of cameras involved in the calculation, f represents the focal length of the camera, d_n represents the depth of the Gaussian sphere observed in the camera cone to the camera's optical center, and if neither is observed, the sampling frequency $v = 0$.

The low-frequency filter can be interpreted as a convolution operation between a low-frequency 3D Gaussian and another Gaussian with a smoothing covariance E, defined as

$$E = \frac{kI}{v} \tag{26}$$

Here, I is the identity matrix and k is a hyperparameter that controls the strength of the filter.

When a low-pass filter is introduced, the scale of the 3D Gauss is expressed in actual rendering as

$$s_f = \sqrt[2]{s^2 + \left(\frac{k}{v}\right)^2 I} \tag{27}$$

4. Main Results.

4.1. Experimental setup.

4.1.1. *Datasets.* The OpenLORIS dataset, released at IROS 2019, provides synchronized RGB-D (30 Hz) and IMU data (gyroscope 400 Hz, accelerometer 250 Hz) from ground robots, with ground truth from OptiTrack. It is suitable for evaluating visual-inertial tracking. We use the *office* sequences. The TUM-RGBD dataset offers synchronized RGB-D and IMU data with motion capture ground truth in static and dynamic scenes, making it ideal for benchmarking RGB-D SLAM and visual-inertial odometry. We select sequences containing both RGB-D and IMU data.

4.1.2. *Evaluation metrics.* In this paper, Absolute Trajectory Error (ATE) [cm] is employed to assess pose estimation accuracy, while Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) are used to evaluate the quality of image rendering.

ATE measures the Root Mean Square Error (RMSE) between the estimated trajectory (after alignment) and the ground-truth trajectory:

$$ATE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|(Rp_i + t) - p_i^{gt}\right\|^2} \tag{28}$$

where p_i is the estimated camera position, p_i^{gt} is the ground-truth position, and R, t are the alignment parameters obtained via least-squares transformation. A lower ATE indicates more accurate trajectory estimation.

For rendering quality, we compute three widely used image quality metrics.

PSNR is defined as

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right), \quad MSE = \frac{1}{N}\sum_{i=1}^{N}\left(I_i - I_i^{gt}\right)^2 \tag{29}$$

where I_i and I_i^{gt} denote the pixel values of the rendered and ground-truth images, and MAX is the maximum possible pixel value. Higher PSNR indicates better fidelity.

SSIM measures structural similarity and is given by

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{30}$$

where μ_x, μ_y are the mean values, σ_x^2, σ_y^2 are the variances, and σ_{xy} is the covariance of images x and y. A higher SSIM indicates better perceptual quality.

LPIPS evaluates perceptual similarity by comparing deep features extracted from a pre-trained neural network:

$$LPIPS(x, y) = \sum_{l}\frac{1}{H_l W_l}\sum_{h,w}\left\|w_l \odot (\phi_l(x)_{hw} - \phi_l(y)_{hw})\right\|^2 \tag{31}$$

where $\phi_l(\cdot)$ denotes the activation of layer l, H_l, W_l are the feature map dimensions, and w_l are learned weights. A lower LPIPS score corresponds to higher perceptual similarity.

During evaluation, we compute the average of each metric over all frames in the sequence.

4.1.3. *Baseline methods.* For comparison, we select the open-source algorithms MM3DGS and MonoGS as baselines. MM3DGS is a multi-modal visual-inertial method that prioritizes real-time performance by directly using IMU data as a positional prior without tight coupling. However, it lacks explicit gravity and velocity initialization, relying on an assumed gravity direction, which limits robustness and may degrade accuracy under incorrect priors. MonoGS is a purely visual SLAM method that does not use IMU data and thus achieves lower accuracy than MM3DGS under ideal conditions, but it can be more robust when IMU priors are unreliable.

4.2. **Evaluation of experimental results.**

4.2.1. *Evaluation of tracking.* To verify the effectiveness of the proposed improvements in IMU data utilization, we compared the positioning accuracy of MM3DGS using IMU fusion and its pure visual variant. As shown in Table 1, MM3DGS with IMU data performed

slightly worse than the pure visual version on the "office" sequences due to the absence of a reliable initialization module and the reliance on a potentially inaccurate gravity prior. After integrating our gravity-aligned initialization module into MM3DGS, the pose estimation accuracy was significantly improved, surpassing even the pure visual baseline. These results confirm the necessity and effectiveness of the proposed initialization strategy. Furthermore, although MonoGS generally lags behind MM3DGS in performance, our enhanced system, which incorporates a tightly coupled visual-inertial optimization module, achieves superior accuracy compared to even the improved MM3DGS, further demonstrating the advantage of our tightly coupled design.

TABLE 1. Comparison of tracking performance on *OpenLORIS* (ATE RMSE [cm])

Methods	**office1-1**	**office1-2**	**office1-3**	**office1-4**	**office1-5**	**office1-6**	**office1-7**
MM3DGS (RGBD only)	7.741	12.83	17.42	19.18	25.23	11.61	33.21
MM3DGS (RGBD+IMU)	7.765	13.05	17.46	19.37	25.30	11.82	33.25
MM3DGS (+ours)	7.732	12.72	17.38	19.10	25.12	11.54	33.17
MonoGS (RGBD only)	8.126	13.21	17.58	19.91	25.99	12.02	33.46
MonoGS (+ours)	6.954	12.27	16.13	18.95	24.61	11.33	31.51
MTCGS (ours)	**6.941**	**12.02**	**15.81**	**18.33**	**24.49**	**11.06**	**31.17**

4.2.2. *Evaluation of mapping.* In this work, we evaluate the quantitative mapping performance of our proposed method on the OpenLORIS and TUM-RGBD datasets, with results summarized in Tables 2 and 3. For evaluation, we use three widely adopted image rendering metrics: PSNR, SSIM, and LPIPS. Our method, MTCGS, consistently outperforms existing approaches across all metrics, achieving the best overall results.

TABLE 2. Comparison of mapping performance on *OpenLORIS*

Methods	**Metrics**	**office1-1**	**office1-2**	**office1-3**	**office1-4**	**office1-5**	**office1-6**	**office1-7**
MM3DGS	PSNR↑	23.90	21.98	26.01	23.91	22.02	21.74	21.02
	SSIM↑	0.801	0.762	0.826	0.849	0.569	0.758	0.578
	LPIPS↓	0.292	0.314	0.285	0.280	0.462	0.318	0.482
MonoGS	PSNR↑	26.85	30.86	30.65	26.85	29.58	25.62	23.33
	SSIM↑	0.884	0.933	0.918	0.895	0.801	0.746	0.842
	LPIPS↓	0.169	0.146	0.179	0.154	0.406	0.230	0.293
Ours	PSNR↑	**28.04**	**31.95**	**31.45**	**28.86**	**30.87**	**26.34**	**24.61**
	SSIM↑	**0.891**	**0.958**	**0.930**	**0.910**	**0.824**	**0.761**	**0.869**
	LPIPS↓	**0.176**	**0.129**	**0.167**	**0.146**	**0.395**	**0.221**	**0.280**

Both MonoGS and MM3DGS rely on the 3DGS framework for mapping, but fail to fully exploit both current and historical keyframe information, and show limited robustness to artifacts in high-frequency textures, leading to suboptimal reconstruction quality.

On the OpenLORIS dataset, we set the number of mapping iterations per keyframe to 60. Our method achieves the highest performance across all three metrics, demonstrating

TABLE 3. Comparison of rendering performance on *TUM-RGBD*

Methods	Metrics	fr1/desk	fr2/xyz	fr3/office
MM3DGS	PSNR↑	17.03	18.35	19.14
	SSIM↑	0.578	0.612	0.778
	LPIPS↓	0.482	0.357	0.296
MonoGS	PSNR↑	23.43	24.70	25.57
	SSIM↑	0.782	0.794	0.852
	LPIPS↓	0.242	0.216	0.192
Ours	PSNR↑	**24.52**	**25.34**	**27.12**
	SSIM↑	**0.801**	**0.830**	**0.894**
	LPIPS↓	**0.242**	**0.216**	**0.181**

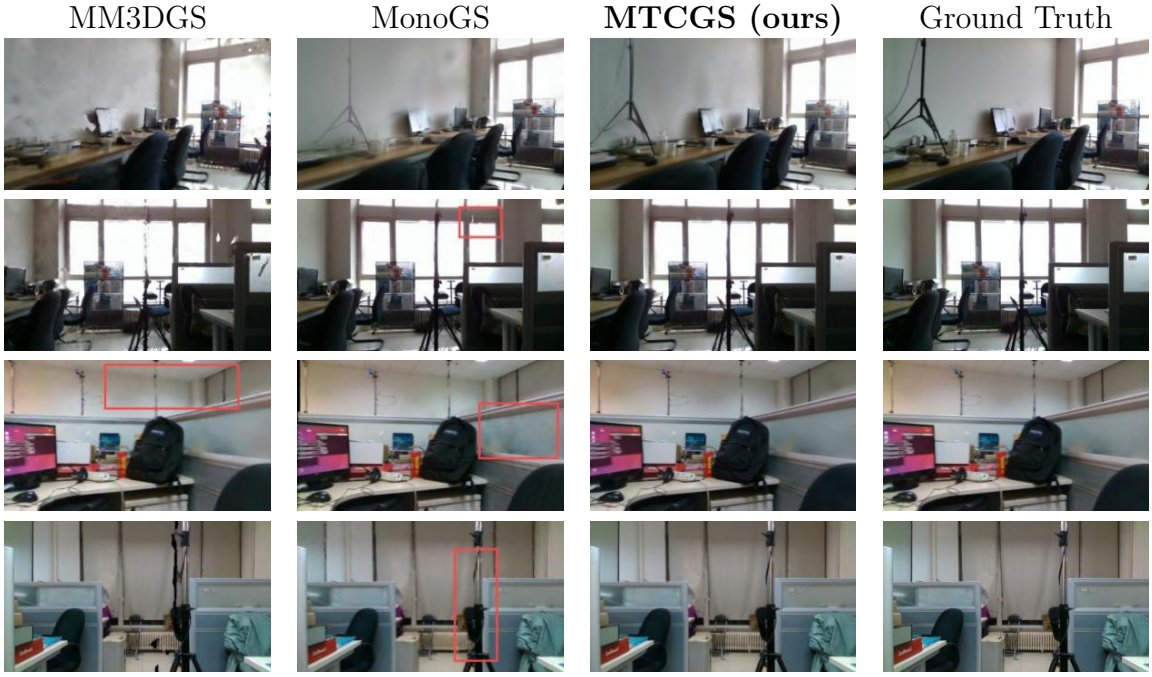

FIGURE 2. Rendering examples on *OpenLORIS*

its effectiveness. On the TUM-RGBD dataset, results are averaged over three representative sequences: fr1/desk, fr2/xyz, and fr3/office. Due to the lower image resolution in TUM-RGBD, the overall mapping quality is somewhat reduced compared to OpenLORIS. Nonetheless, our method still achieves the best performance among all approaches.

Furthermore, since TUM-RGBD does not provide IMU data, these results also indicate that our mapping strategy can effectively improve mapping quality even without additional motion information. Overall, the experimental results on both datasets demonstrate the robustness and effectiveness of our proposed approach.

Figure 2 compares the qualitative mapping results on the OpenLORIS dataset. MM-3DGS shows severe ceiling aliasing and texture loss, while MonoGS exhibits artifacts and rough reconstruction on frosted glass. In contrast, our method effectively suppresses aliasing and artifacts, achieving more robust and accurate texture reconstruction.

5. **Conclusion.** We propose MTCGS, a multi-modal 3D Gaussians SLAM algorithm. The pre-integration module and initialization module are implemented to solve the robustness problem in the process of coupling 3D Gaussian SLAM with inertial data. In

order to solve the artifact problem in the process of constructing 3D Gaussian SLAM, an adaptive optimization window and an improved 3D Gaussian smoothing filter are designed in this paper. This paper evaluates our framework on the OpenLORIS and TUM-RGBD datasets. Compared with the most advanced baselines, the IMU fusion part in this paper is more robust, and the mapping module can effectively handle high-frequency artifacts. Due to the low price of IMU sensors, in future practical applications, MTCGS can be further applied in robotics, augmented reality and other fields.

REFERENCES

[1] E. Sandström, Y. Li, L. Van Gool and M. R. Oswald, Point-SLAM: Dense neural point cloud-based SLAM, *IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, pp.18387-18398, 2023.

[2] F. Ruetz, E. Hernández, M. Pfeiffer, H. Oleynikova, M. Cox and T. Lowe, OVPC Mesh: 3D free-space representation for local ground vehicle navigation, *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, pp.8648-8654, 2019.

[3] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu and G. Zhang, Vox-Fusion: Dense tracking and mapping with voxel-based neural implicit representation, *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Singapore, pp.499-507, 2022.

[4] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi and R. Ng, NeRF: Representing scenes as neural radiance fields for view synthesis, *Communications of the ACM*, vol.65, no.1, pp.99-106, 2021.

[5] B. Kerbl, G. Kopanas, T. Leimkühler and G. Drettakis, 3D Gaussian Splatting for real-time radiance field rendering, *ACM Transactions on Graphics*, vol.42, no.4, 2023.

[6] E. Sucar, S. Liu, J. Ortiz and A. J. Davison, iMAP: Implicit mapping and positioning in real-time, *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp.6229-6238, 2021.

[7] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald and M. Pollefeys, NICE-SLAM: Neural implicit scalable encoding for SLAM, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.12786-12796, 2022.

[8] M. M. Johari, C. Carta and F. Fleuret, ESLAM: Efficient dense SLAM system based on hybrid representation of signed distance fields, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.17408-17419, 2023.

[9] Z. Peng, T. Shao, Y. Liu, J. Zhou, Y. Yang, J. Wang and K. Zhou, RTG-SLAM: Real-time 3D reconstruction at scale using Gaussian Splatting, *ACM SIGGRAPH 2024 Conference Papers*, 2024.

[10] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan and J. Luiten, SplaTAM: Splat track & Map 3D Gaussians for dense RGB-D SLAM, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.21357-21366, 2024.

[11] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang and X. Li, GS-SLAM: Dense visual SLAM with 3D Gaussian Splatting, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.19595-19604, 2024.

[12] D. Lisus, C. Holmes and S. Waslander, Towards open world NeRF-based SLAM, *2023 20th Conference on Robots and Vision (CRV)*, pp.37-44, 2023.

[13] L. C. Sun, N. P. Bhatt, J. C. Liu, Z. Fan, Z. Wang, T. E. Humphreys and U. Topcu, MM3DGS SLAM: Multi-modal 3D Gaussian Splatting for SLAM using vision, depth, and inertial measurements, *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.10159-10166, 2024.

[14] A. I. Mourikis and S. I. Roumeliotis, A multi-state constraint Kalman filter for vision-aided inertial navigation, *Proc. of 2007 IEEE International Conference on Robotics and Automation*, pp.3565-3572, 2007.

[15] T. Qin, P. Li and S. Shen, VINS-Mono: A robust and versatile monocular visual-inertial state estimator, *IEEE Transactions on Robotics*, vol.34, no.4, pp.1004-1020, 2018.

[16] V. Yugay, Y. Li, T. Gevers and M. R. Oswald, Gaussian-SLAM: Photo-realistic dense SLAM with Gaussian splatting, *arXiv Preprint*, arXiv: 2312.10070, 2023.

[17] Z. Teed and J. Deng, DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras, *Advances in Neural Information Processing Systems,* vol.34, pp.16558-16569, 2021.

[18] R. Mur-Artal and J. D. Tardós, ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras, *IEEE Transactions on Robotics*, vol.33, no.5, pp.1255-1262, 2017.

[19] J. Engel, V. Koltun and D. Cremers, Direct sparse odometry, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.40, no.3, pp.611-625, 2017.
[20] C. Kerl, J. Sturm and D. Cremers, Dense visual SLAM for RGB-D cameras, *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2100-2106, 2013.
[21] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi and C. Theobalt, BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration, *ACM Transactions on Graphics (ToG)*, vol.36, no.4, 2017.
[22] M. Hsiao, E. Westman, G. Zhang and M. Kaess, Keyframe-based dense planar SLAM, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp.5110-5117, 2017.
[23] A. Kasyanov, F. Engelmann, J. Stückler and B. Leibe, Keyframe-based visual-inertial online SLAM with relocalization, *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.6662-6669, 2017.
[24] T. Schops, T. Sattler and M. Pollefeys, BAD SLAM: Bundle adjusted direct RGB-D SLAM, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.134-144, 2019.
[25] D. R. Canelhas, T. Stoyanov and A. J. Lilienthal, SDF Tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images, *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.3671-3676, 2013.
[26] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht and A. Kanazawa, Plenoxels: Radiance fields without neural networks, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.5501-5510, 2022.
[27] Z. Yu, A. Chen, B. Huang, T. Sattler and A. Geiger, Mip-Splatting: Alias-free 3D Gaussian Splatting, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.19447-19456, 2024.
[28] H. Matsuki, R. Murai, P. H. J. Kelly and A. J. Davison, Gaussian Splatting SLAM, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.18039-18048, 2024.
[29] C. Forster, L. Carlone, F. Dellaert and D. Scaramuzza, On-manifold preintegration for real-time visual-inertial odometry, *IEEE Transactions on Robotics*, vol.33, no.1, pp.1-21, 2016.
[30] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart and P. Furgale, Keyframe-based visual-inertial odometry using nonlinear optimization, *The International Journal of Robotics Research*, vol.34, no.3, pp.314-334, 2015.

Author Biography

Fenghua Wang graduated from Xi'an Jiaotong University, China, with a Ph.D. degree in Engineering in June 2009 and is a member of the Chinese Computer Society. He is currently employed at the College of Computer Science and Technology, China University of Petroleum (East China), China, where he works at the Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software. His research interests include image processing and computer vision, AI edge computing, embedded software development, etc.

Yachao Hu graduated with a Bachelor's degree from Zhengzhou University, China, in 2023. He is currently pursuing a Master's degree in Software Engineering at the College of Computer Science and Technology, China University of Petroleum (East China), China, where he studies at the Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software. His research interests include SLAM and computer vision.

Zhicheng Xu graduated with a Bachelor's degree from Zhengzhou University, China, in 2020. He is currently pursuing a Master's degree in Software Engineering at the College of Computer Science and Technology, China University of Petroleum (East China), China, where he studies at the Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software. His research interests include SLAM and visual localization.

Yixin Zhang is currently pursuing a Bachelor's degree in Software Engineering at the College of Computer Science and Technology, China University of Petroleum (East China), China. His interests include machine learning and computer vision.