remote sensing

MDPI

Article

Geo-Consistent Centralized Multi-UAV Gaussian SLAM for Incremental Orthophoto Generation

Xiao Zhang [1,2,†], Shuaixin Li [2,3,†], Hongbin Dong [1], Xiaozhou Zhu [2,3], Haoxin Zhang [2,4] and Baosong Deng [2,3,*]

[1] College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China; zhangxiao516@hrbeu.edu.cn (X.Z.); donghongbin@hrbeu.edu.cn (H.D.)
[2] Defense Innovation Institute, Chinese Academy of Military Science, Beijing 100071, China; lsx_navigation@sina.com (S.L.); zhuxiaozhou.niidt@gmail.com (X.Z.); zhanghx97@mail2.sysu.edu.cn (H.Z.)
[3] Intelligent Game and Decision Laboratory, Beijing 100071, China
[4] College of Systems Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China
* Correspondence: dbs@nudt.edu.cn
† These authors contributed equally to this work.

Highlights

What are the main findings?

- A geo-consistent centralized multi-UAV Gaussian SLAM framework is proposed, combining GNSS-prior alignment with inter-agent visual loop refinement to fuse multiple UAV observations into a unified global Gaussian map.
- Plane-guided collaborative densification and visibility-aware adaptive pruning improve mapping quality in under-constrained regions while controlling Gaussian redundancy, memory consumption, and computational cost.

What are the implications of the main findings?

- The proposed framework enables online incremental and geo-referenced orthophoto generation from distributed observations captured by multiple consumer-grade UAVs.
- The two Gaussian map maintenance modules support efficient and scalable mapping over extended areas, making the framework suitable for time-sensitive aerial mapping applications.

Abstract

Online incremental orthophoto generation with multiple unmanned aerial vehicles (UAVs) remains challenging, as it requires accurate, efficient, and scalable mapping from distributed aerial observations. In this paper, we present a centralized GNSS-assisted multi-UAV 3D Gaussian Splatting SLAM framework for online incremental orthophoto mapping. Each UAV independently performs visual odometry to build local submaps, which are first aligned into a unified global coordinate system using GNSS constraints and further refined via inter-agent visual loop closures for improved cross-agent consistency. To enable scalable and high-quality mapping, we introduce two complementary Gaussian map maintenance modules: plane-guided grid-based collaborative densification, which improves mapping quality and accelerates convergence under multi-UAV conditions, and visibility-aware adaptive pruning, which effectively controls redundancy and memory usage. These components allow efficient joint optimization within a unified Gaussian representation. Experiments on multiple aerial datasets using video-derived image frames captured by consumer-grade UAV cameras demonstrate that the proposed system provides a favorable trade-off between geo-consistency, visual fidelity, and efficiency compared with existing methods. Quantitatively, the proposed method achieves a GCP RMSE of 2.32 m, completes multi-UAV orthophoto generation within 3.3–5.5 min, and reduces the

Check for updates

Academic Editor: Shuanggen Jin

Received: 2 June 2026
Revised: 1 August 2026
Accepted: 17 August 2026
Published: 19 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

total mapping time by approximately 35–55% compared with the corresponding single-UAV setting, while supporting online tracking and incremental orthophoto updates with bounded latency and memory consumption.

Keywords: multi-UAV mapping; 3D Gaussian Splatting; visual SLAM; orthophoto generation; online mapping

1. Introduction

Unmanned aerial vehicles (UAVs) have become an essential sensing platform for remote sensing and photogrammetric applications, including terrain mapping, infrastructure inspection, disaster response, and environmental monitoring [1]. Compared with traditional manned aerial surveys, UAV systems offer flexible deployment, low operational cost, and rapid data acquisition capabilities, making them particularly suitable for mapping complex or inaccessible environments [2,3]. However, generating orthomosaics using a single UAV remains inherently constrained by limited flight endurance and restricted spatial coverage, which limit mapping efficiency and scalability as the surveyed area and sequence length increase. These limitations have motivated increasing interest in collaborative multi-UAV mapping, where multiple aerial agents simultaneously explore different regions to improve coverage efficiency, robustness, and operational scalability [4]. Despite its promise, practical multi-UAV orthomosaic generation remains challenging due to the need for geo-consistent alignment and efficient incremental fusion across overlapping aerial observations, as well as scalable scene representation for extended-area mapping. An overview of the multi-UAV online orthomosaic mapping pipeline is illustrated in Figure 1. These challenges highlight the need for unified and geo-consistent mapping frameworks capable of integrating multi-agent observations in real time.

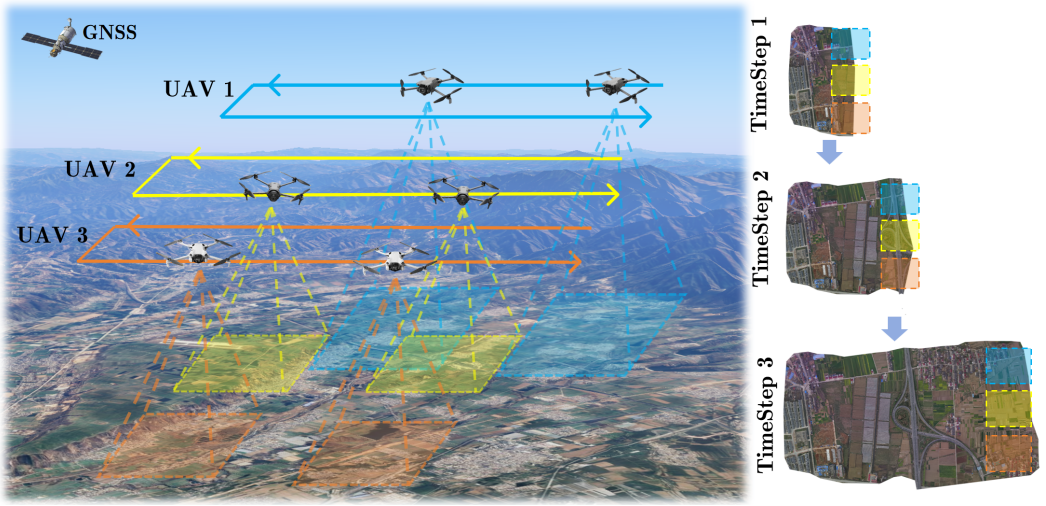


Figure 1. Illustration of online multi-UAV orthomosaic generation. Multiple UAVs equipped with GNSS capture overlapping aerial imagery from different viewpoints, while the orthomosaic is incrementally constructed over time as new observations are integrated.

Conventional orthomosaic generation pipelines predominantly rely on structure-from-motion (SfM) [5] and multi-view stereo (MVS) [6], which generally require the complete image set to be collected before performing global feature matching, bundle adjustment, dense reconstruction, and orthorectification. These offline batch-processing pipelines are well suited to post-mission applications in which high geometric accuracy and reconstruc-

tion completeness are the primary objectives [1]. However, they cannot continuously provide updated spatial information during data acquisition, and their processing latency generally increases with the number of images and the surveyed area [7–9]. Consequently, they may be unsuitable for time-sensitive applications, such as disaster response, emergency inspection, rapid environmental assessment, and dynamic situational awareness, where map information is required before the UAV mission is completed [4].

To enable online mapping, recent studies have explored SLAM-based orthomosaic generation by incrementally estimating camera trajectories and integrating incoming images into a global mosaic [4]. Unlike offline batch processing, online incremental mapping progressively updates the trajectory and map as new observations arrive, thereby enabling early visualization and continuous spatial feedback during UAV operation. Although these approaches reduce processing latency, many of them employ sparse geometric representations or simplified projection models, such as planar assumptions or local homography estimation, which may degrade orthorectification quality in scenes with significant elevation variations or complex structures [10–13]. Moreover, most existing systems focus on single-agent mapping and provide limited support for cross-agent coordination and joint map optimization. Extending them to multi-UAV scenarios therefore requires consistent alignment, fusion, and optimization of observations from multiple agents. This motivates a unified representation that supports efficient incremental fusion, consistent geometric modeling, and scalable multi-agent optimization.

Recently, 3D Gaussian Splatting (3DGS) has attracted increasing attention due to its ability to jointly model geometry and appearance with efficient differentiable rendering [14]. Building upon this representation, 3DGS-based SLAM [15–18] integrates dense scene modeling into the SLAM pipeline, enabling simultaneous pose estimation and photorealistic mapping. Compared with conventional SLAM methods based on sparse features or local geometric approximations, 3DGS-SLAM provides a denser and more visually consistent scene representation, making it a promising basis for UAV orthophoto generation. By optimizing a Gaussian representation and rendering it orthographically, such methods [19,20] can alleviate distortions caused by local homography assumptions and perspective variations in stitching-based approaches. However, applying 3DGS-SLAM to online orthophoto mapping remains challenging. Incremental reconstruction is sensitive to weakly constrained regions, such as texture-poor or reflective surfaces and trajectory boundaries with insufficient multi-view observations, where conventional densification may lack sufficient geometric guidance. Meanwhile, the continuous growth of Gaussian primitives increases memory consumption and optimization cost. These challenges become more significant in multi-UAV scenarios, where observations from different agents must be consistently aligned and fused into a scalable shared representation.

To address these challenges, we propose a geo-consistent centralized multi-UAV Gaussian SLAM framework for online incremental orthomosaic generation. In our system, each UAV independently performs visual odometry to estimate local trajectories and construct sparse submaps, which are progressively aligned into a shared geo-referenced coordinate frame using consumer-grade onboard GNSS measurements. GNSS provides coarse metric and geographic priors for estimating the initial agent-level Sim(3) transformations, while inter-agent visual loop closures further refine residual alignment errors and improve cross-agent consistency. The proposed framework assumes the availability of valid GNSS observations during global alignment. Instead of maintaining multiple independent Gaussian maps, all observations from different agents are jointly optimized within a single global Gaussian representation, enabling consistent geometry and appearance modeling across overlapping flight regions. To achieve scalable and high-quality orthomosaic generation, we introduce two complementary Gaussian map maintenance modules: plane-guided

grid-based collaborative densification and visibility-aware adaptive pruning. The former leverages planar geometric priors to expand Gaussian primitives in under-constrained regions, while the latter controls redundancy and memory growth by removing persistently low-contribution Gaussians, enabling efficient and scalable long-term incremental mapping. These components together establish a unified and scalable multi-UAV Gaussian mapping framework for online incremental orthomosaic generation in aerial environments. The main contributions of this work are summarized as follows:

- We propose a geo-consistent centralized multi-UAV Gaussian SLAM framework for online incremental orthomosaic generation, where multiple agents are jointly optimized within a unified global Gaussian representation through GNSS-assisted alignment and inter-agent visual loop closure refinement.
- We develop two complementary Gaussian map maintenance modules, namely, plane-guided grid-based collaborative densification and visibility-aware adaptive pruning, which improve orthomosaic quality in under-constrained regions while reducing redundant Gaussians and controlling model size and memory growth during incremental mapping.
- Extensive experiments on UAV aerial datasets demonstrate that the proposed system achieves competitive orthorectification accuracy, photometric consistency, and scalability compared with existing orthophoto generation pipelines, while maintaining real-time tracking and online incremental orthomosaic updates.

2. Related Works

2.1. SLAM and Multi-Agent SLAM

Visual and visual–inertial SLAM have been extensively studied for online localization and mapping on single-UAV platforms. ORB-SLAM2 [21] and ORB-SLAM3 [22] employ feature-based tracking and bundle adjustment, DSO [23] estimates camera motion through direct photometric optimization, and VINS-Mono [24] tightly couples visual and inertial measurements. These methods mainly estimate camera trajectories in a local coordinate frame and generally do not directly provide an absolute geographic reference [25]. GNSS-integrated systems, such as VINS-Fusion [26], GVINS [27], and GIVL-SLAM [28], incorporate GNSS observations as frame-level or state-level constraints within continuous trajectory estimation to recover globally referenced poses. These GNSS-integrated systems mainly focus on globally referenced state estimation for a single platform, rather than the alignment of independently initialized multi-UAV submaps. In our framework, synchronized visual and GNSS positions are used to estimate an agent-level Sim(3) transformation for global submap alignment, which is further refined through inter-agent visual loop closures.

Multi-agent SLAM has attracted increasing attention for collaborative localization and large-area mapping. CCM-SLAM [29] adopts a client–server architecture for inter-robot loop closure and centralized optimization, while COVINS [30] maintains a shared keyframe graph for collaborative localization. Kimera-Multi [31] and MA-LIO [32] further extend collaborative mapping to visual–inertial and LiDAR-based systems. However, these approaches mainly focus on collaborative pose estimation and sparse or geometric map optimization. In contrast, the proposed framework aligns multi-UAV submaps within a shared geographic frame and jointly integrates their observations into a unified dense Gaussian representation for incremental aerial orthophoto generation.

2.2. 3D Gaussian Splatting and Gaussian-Based SLAM

The introduction of 3D Gaussian Splatting (3DGS) has significantly advanced dense and photorealistic scene modeling by representing environments using anisotropic Gaus-

sian primitives optimized through differentiable rasterization [14]. Compared with NeRF-based [33] implicit representations, 3DGS achieves orders-of-magnitude faster rendering while preserving high visual fidelity.

Following its emergence, numerous studies have focused on improving the efficiency of 3DGS representations. PUP 3D-GS [34] introduces a principled sensitivity-based pruning strategy to remove redundant Gaussians while preserving visual fidelity. Speedy-Splat [35] improves rendering and training efficiency through sparse pixel and primitive processing, while EfficientGS [36] combines selective densification, Gaussian pruning, and sparse spherical harmonics to reduce model size and computational overhead. These approaches significantly improve the efficiency and compactness of 3DGS, but mainly operate on pre-collected static scenes rather than online mapping. In parallel, large-scale extensions such as DOGS [37] and the CityGaussian series [38,39] employ distributed or block-wise optimization and hierarchical level-of-detail representations to support large-scale scene reconstruction and rendering. However, these methods still rely on scene-level offline optimization and do not explicitly address online camera tracking and incremental fusion into a unified Gaussian representation.

Building on this representation, a growing body of work integrates 3DGS into SLAM pipelines for joint pose estimation and dense mapping. Representative systems include GS-SLAM [40], Photo-SLAM [16], SplaTAM [18], Gaussian-SLAM [17], and Compact-GSSLAM [41], which adopt 3D Gaussians as the core map representation and estimate camera poses by minimizing photometric residuals between rendered and observed images, while incrementally updating Gaussian parameters via gradient-based optimization. Subsequent extensions introduce additional components such as loop closure and external visual odometry [16], ICP-based geometric alignment [42], or multi-modal inputs [43]. More recent variants further explore semantics [44], depth priors [45], and LiDAR–visual fusion [46], highlighting the flexibility of Gaussian primitives across sensing modalities. Despite these advances, existing Gaussian-based SLAM systems are predominantly designed for single-agent scenarios. Directly extending them to multi-UAV settings by maintaining multiple Gaussian submaps leads to duplicated splats, appearance conflicts, and unstable convergence. Moreover, global densification and pruning strategies can easily corrupt previously optimized regions, making scalable and consistent multi-agent Gaussian mapping an open challenge.

2.3. Orthomosaic Generation in Aerial Mapping

Orthomosaic generation methods can generally be categorized into reconstruction-driven and SLAM-driven approaches. Reconstruction-based pipelines typically rely on SfM [5] and MVS [6], as implemented in commercial software such as Pix4DMapper (version 4.8) [9] and Metashape (version 2.0) [7]. These methods reconstruct dense 3D models offline and project them onto orthogonal planes to generate orthophotos. More recent implicit or Gaussian-based approaches, including TDOM-NeRF [20], TorthoGaussian [19], and Ortho-3DGS [47], directly perform orthographic rendering from aerial imagery using neural or Gaussian representations, achieving high visual fidelity. However, these methods require substantial computational resources and long optimization times, rendering them unsuitable for real-time incremental deployment.

In contrast, SLAM-based mosaicking emphasizes online processing. Map2DFusion [10], OpenRealm [12], and Real-Time Increment [11] perform incremental alignment and fusion using keyframes or local map optimization, while GeoROS [48] and EffcientOrtho [13] incorporate GNSS measurements to improve global consistency. Although these approaches enable online stitching, the resulting products are typically pseudo-orthophotos that still exhibit parallax artifacts and facade distortions, especially in areas with large elevation variations.

Importantly, existing SLAM-based orthomosaic systems are almost exclusively designed for single-UAV operation and do not exploit dense Gaussian representations. More importantly, existing SLAM-based orthomosaic methods rely on sparse geometry or planar assumptions, while recent Gaussian-based orthographic approaches remain offline and single-agent. A unified framework that jointly integrates multi-UAV SLAM, dense Gaussian representation, and online orthomosaic generation has not yet been investigated.

3. Materials and Methods

3.1. System Overview: Centralized Multi-UAV Gaussian SLAM Architecture

As illustrated in Figure 2, our system adopts a centralized multi-UAV architecture for geo-consistent Gaussian mapping and online orthomosaic generation. The input consists of video-derived image frames captured by consumer-grade onboard UAV cameras. Each UAV independently runs a visual odometry frontend to estimate local poses and sparse maps, while keyframes, triangulated points, and synchronized GNSS measurements are incrementally provided to the centralized backend for global fusion and optimization.

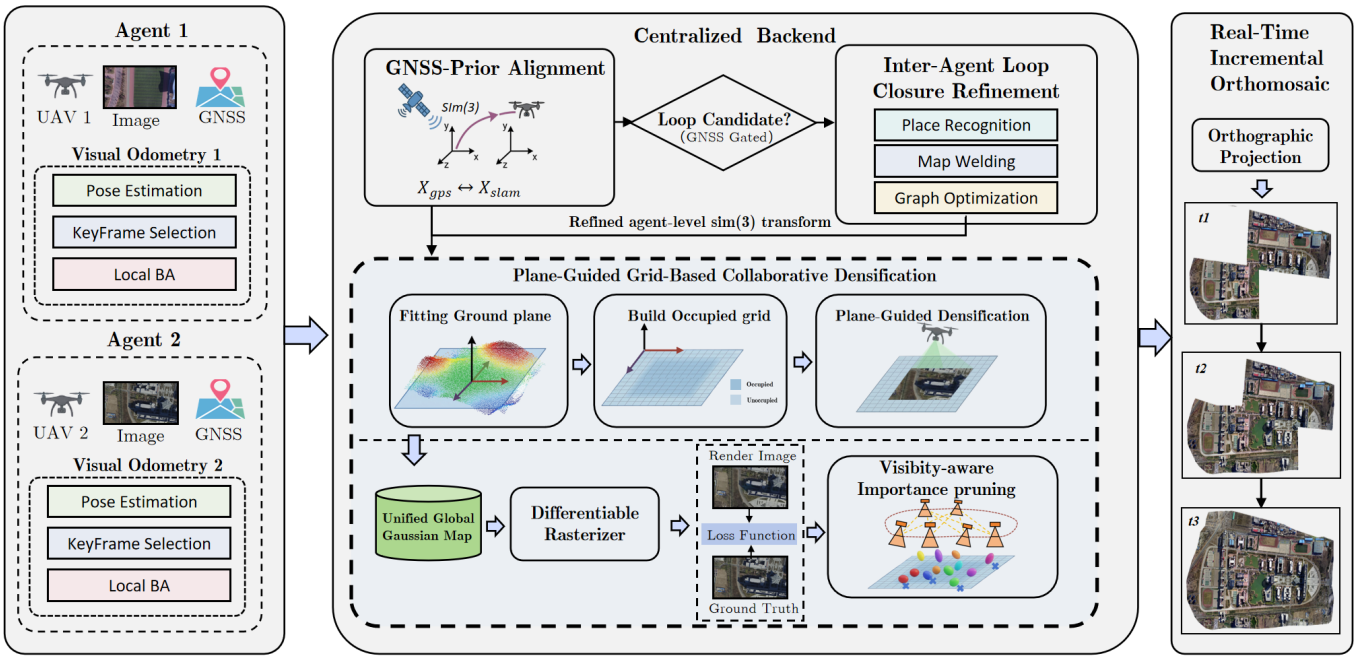


Figure 2. Overview of the proposed centralized multi-UAV Gaussian SLAM framework. UAVs perform GNSS-assisted visual odometry, while a centralized backend ensures consistent multi-agent alignment and refinement. A unified Gaussian map is incrementally optimized, and orthomosaics are generated via orthographic projection.

At the backend, all agent trajectories are progressively aligned into a unified global frame via GNSS-assisted initialization, followed by inter-agent visual loop closure refinement to further refine the relative transformations between agents, enabling robust multi-UAV coordination under large baselines and partial overlap. Aligned observations from all agents are fused into a single global Gaussian representation, where new primitives are initialized from sparse points and incrementally optimized via differentiable rendering. Instead of maintaining independent submaps, all agents collaboratively update a unified model.

To ensure scalability, plane-guided collaborative densification and visibility-aware pruning are applied to regulate Gaussian growth and preserve previously optimized regions. Finally, the optimized Gaussian map is orthographically rendered to produce online incremental orthomosaics.

3.2. GNSS-Prior Multi-UAV Alignment with Visual Loop Refinement

We achieve geo-consistent multi-UAV coordination by combining GNSS-assisted absolute alignment with inter-agent visual loop refinement. As illustrated in Figure 3, GNSS first establishes a shared metric reference frame, enabling the alignment of local SLAM trajectories from multiple agents into a unified global coordinate system. This provides a consistent basis for fusing all observations into a centralized Gaussian map. Subsequently, cross-agent visual loop detection is performed to identify overlapping regions, where reliable correspondences are established to further reduce residual misalignment. In addition, our refinement operates only at the agent level: it updates each agent's global Sim(3) parameters $(s^a, \mathbf{R}^a, \mathbf{t}^a)$ rather than modifying all keyframe poses, thereby preserving an efficient incremental pipeline.

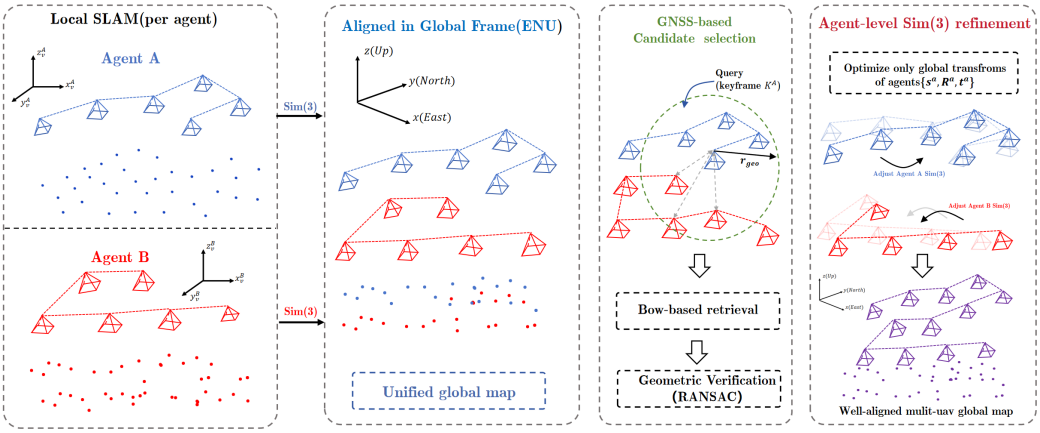


Figure 3. Illustration of the proposed multi-UAV alignment framework. Keyframes are shown as square pyramids and map points as dots, with different colors indicating different UAV agents. The dashed circle represents GNSS-based spatial gating, and dashed lines denote inter-keyframe connections. The final result shows the aligned multi-UAV global map.

3.2.1. GNSS-Based Global Alignment

Each agent a runs monocular SLAM in its local coordinate frame $\mathcal{C}_v$, which lacks absolute scale and global orientation. To establish a geo-referenced mapping, we estimate a similarity transformation $\mathbf{T}^a_{vg} = (s^a, \mathbf{R}^a, \mathbf{t}^a)$ that aligns the visual coordinate system $\mathcal{C}_v$ with the GNSS-aligned global frame $\mathcal{C}_g$. A 3D point $\mathbf{x}^v \in \mathcal{C}_v$ is transformed to the global frame as follows:

$$\mathbf{x}^g = s^a \mathbf{R}^a \mathbf{x}^v + \mathbf{t}^a. \tag{1}$$

The transformation is initialized using a closed-form alignment between corresponding keyframe positions in the two coordinate systems. Specifically, given a set of matched positions $\{\mathbf{x}^v_i, \mathbf{x}^g_i\}_{i=1}^N$, $\mathbf{x}^v_i$ denotes the camera center estimated by visual SLAM, while $\mathbf{x}^g_i$ is the synchronized onboard GNSS position at the corresponding keyframe timestamp, converted into the local ENU frame. We compute the centroids:

$$\bar{\mathbf{x}}^v = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^v_i, \quad \bar{\mathbf{x}}^g = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^g_i, \tag{2}$$

and form the covariance matrix:

$$\mathbf{H} = \sum_{i=1}^{N} (\mathbf{x}^g_i - \bar{\mathbf{x}}^g)(\mathbf{x}^v_i - \bar{\mathbf{x}}^v)^\top. \tag{3}$$

By performing singular value decomposition (SVD) on $\mathbf{H}$, the optimal rotation $\mathbf{R}^a$ is obtained, while the scale s^a and translation $\mathbf{t}^a$ are computed accordingly following a Procrustes alignment formulation. This closed-form initialization provides a coarse estimate of the global Sim(3) transformation.

To further improve robustness under weak vertical excitation and GNSS noise, we optionally regularize the alignment by enforcing consistency between the estimated vertical direction and GNSS-derived altitude trends, as well as validating with an estimated ground-plane normal.

3.2.2. GNSS-Guided Candidate Selection and BoW-Based Matching

Direct cross-agent loop search is expensive and prone to false positives. We exploit the GNSS prior to restrict candidates to geographically plausible overlaps. For a keyframe k from agent a, let $\mathbf{c}_k^{v,a}$ denote its camera center in the visual SLAM coordinate frame $\mathcal{C}_v^a$. Its corresponding position in the global frame $\mathcal{C}_g$ is given by:

$$\mathbf{c}_k^{g,a} \;=\; s^a \mathbf{R}^a \mathbf{c}_k^{v,a} + \mathbf{t}^a. \tag{4}$$

For another agent $b \neq a$, we restrict candidate keyframes to those within a spatial radius r_{geo} in the global frame:

$$\mathcal{N}_{\text{geo}}(k) \;=\; \Big\{ l \in \mathcal{K}^b \;\Big|\; \big\|\mathbf{c}_k^{g,a} - \mathbf{c}_l^{g,b}\big\|_2 < r_{\text{geo}} \Big\}. \tag{5}$$

Consumer-grade single-point GNSS typically exhibits horizontal positioning error of approximately 2–5 m, respectively, under open-sky conditions [49]. Accordingly, r_{geo} is set to 20 m as a permissive threshold to accommodate typical GNSS uncertainty and residual initial alignment errors. This GNSS-initialized gating strategy reduces the search space while retaining geographically plausible overlap candidates under typical GNSS errors.

After this spatially constrained set, we perform appearance-based place recognition. Within $\mathcal{N}_{\text{geo}}(k)$, we perform inter-agent place recognition using ORB Bag-of-Words (BoW). Given a query keyframe k from agent a, we retrieve the top-K most similar keyframes from agent b and obtain $\mathcal{N}_{\text{bow}}(k) \subseteq \mathcal{N}_{\text{geo}}(k)$. Each candidate pair (k, l), consisting of a query keyframe k from agent a and a retrieved keyframe l from agent b, is then validated by robust geometric verification using feature correspondences and associated map-point matches.

Successful verification yields a set of matched 3D point correspondences $\{(\mathbf{x}_i^{v,a}, \mathbf{x}_i^{v,b})\}$ defined in the respective local SLAM coordinate frames $\mathcal{C}_v^a$ and $\mathcal{C}_v^b$. These correspondences are used as anchor constraints for subsequent inter-agent alignment.

3.2.3. Agent-Level Sim(3) Refinement

Let $\mathcal{E}$ denote the set of verified inter-agent constraints. Instead of refining all keyframe poses, we optimize only the agent-level similarity transformations $\{(s^a, \mathbf{R}^a, \mathbf{t}^a)\}$ so that corresponding points from different agents are consistent in the global frame. Specifically, given matched anchor pairs $\{(\mathbf{x}_i^{v,a}, \mathbf{x}_i^{v,b})\}$ for each $(a, b) \in \mathcal{E}$, we minimize the following objective:

$$\min_{\{s^a, \mathbf{R}^a, \mathbf{t}^a\}} \sum_{(a,b)\in\mathcal{E}} \sum_i \rho\Big(\Big\| s^a \mathbf{R}^a \mathbf{x}_i^{v,a} + \mathbf{t}^a - \Big(s^b \mathbf{R}^b \mathbf{x}_i^{v,b} + \mathbf{t}^b \Big) \Big\|_2 \Big) \tag{6}$$

where $\rho(\cdot)$ is a robust loss function, and one agent is fixed to eliminate gauge freedom. The refinement is performed incrementally with change-magnitude gating to prevent over-correction.

Overall, the proposed GNSS-prior and BoW-gated coordination framework enables robust and scalable multi-UAV alignment by jointly leveraging global geo-referencing and

visual consistency. GNSS provides a reliable global reference and spatial gating, while inter-agent visual correspondences further refine relative alignment in overlapping regions. Compared with GNSS-only alignment methods, the proposed approach effectively reduces residual misalignment caused by GNSS noise.

3.3. Plane-Guided Collaborative Densification

Multi-UAV orthomosaic mapping suffers from highly unbalanced supervision: large ground regions are often under-constrained due to limited baselines, motion blur, repetitive patterns, and weak parallax. In such cases, conventional densification fails to provide sufficient geometric support for efficient Gaussian optimization, limiting convergence speed and hindering online incremental orthomosaic generation. Inspired by orthomosaic image stitching, we propose a plane-guided and grid-based collaborative densification strategy, as illustrated in Figure 4.

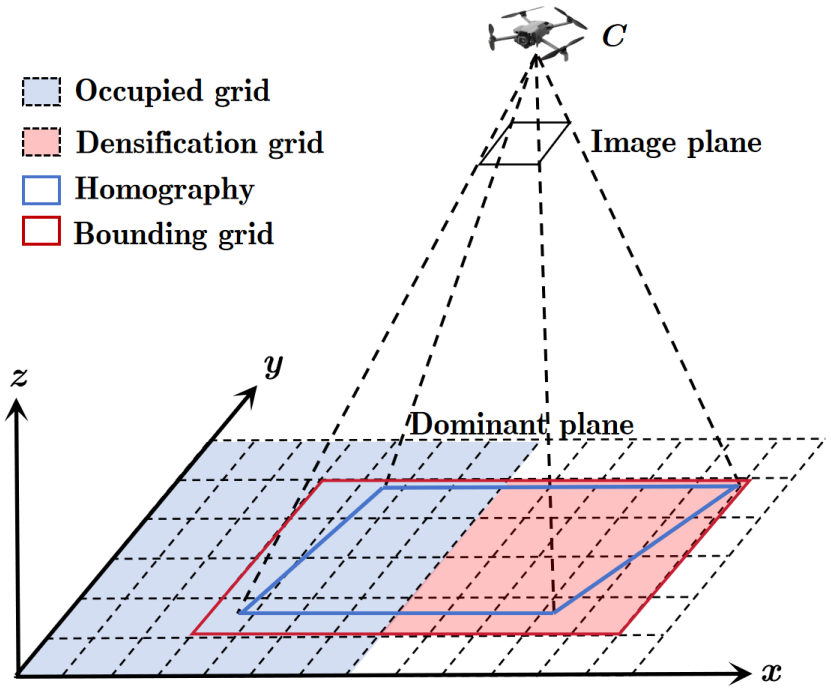


Figure 4. Illustration of the plane-guided collaborative densification strategy. A dominant ground plane is estimated to construct an occupied grid, within which homography-guided densification is performed to expand Gaussian representation in under-constrained regions.

The method uses a dominant global ground plane to provide geometry-aware Gaussian initialization in weakly constrained regions. A planar grid serves as persistent spatial memory to coordinate densification across agents, while a gradient-based signal triggers re-densification in non-converged areas. This design improves reconstruction quality and accelerates convergence under multi-UAV conditions.

3.3.1. Global Plane and Grid-Based Coverage Control

After GNSS–SLAM alignment, each agent a provides a Sim(3) transformation $(s^a, \mathbf{R}^a, \mathbf{t}^a)$ from its local visual coordinate frame $\mathcal{C}_v^a$ to the global frame $\mathcal{C}_g$. Ground-support points from all agents are unified in the global frame as follows:

$$\mathbf{x}^g = s^a \mathbf{R}^a \mathbf{x}^v + \mathbf{t}^a, \tag{7}$$

where $\mathbf{x}^v \in \mathcal{C}_v^a$ and $\mathbf{x}^g \in \mathcal{C}_g$. Following the plane estimation strategy in EfficientOrtho [13], we robustly estimate a dominant ground plane from the unified ground-support points. The plane parameters are then refined from the inlier set to obtain the following:

$$\Pi: \quad \mathbf{n}^\top \mathbf{x}^g = d, \qquad d = \mathbf{n}^\top \mathbf{p}_0, \tag{8}$$

where $\mathbf{n}$ is a unit normal and $\mathbf{p}_0$ is a point on the plane. Fitting a common global plane provides a shared geometric reference for collaborative densification across agents.

To facilitate a compact 2D parameterization on the plane, we further define a plane-aligned coordinate frame $\mathbf{R}_\pi \in SO(3)$ whose z-axis is aligned with $\mathbf{n}$.

To prevent uncontrolled growth of Gaussians, we discretize the global plane into a 2D grid with cell size g. For a 3D point $\mathbf{x}$ near the plane, its plane-local coordinates and grid index are computed as

$$\mathbf{q} = \mathbf{R}_\pi^\top (\mathbf{x} - \mathbf{p}_0), \qquad (i, j) = \left(\left\lfloor \frac{q_x}{g} \right\rfloor, \left\lfloor \frac{q_y}{g} \right\rfloor \right). \tag{9}$$

Each cell key (i, j) is stored in a persistent set $\mathcal{C}_{\text{covered}}$, which acts as a lightweight spatial memory indicating which planar regions have already been densified. This mechanism enforces that densification is primarily directed to previously uncovered planar areas, which is crucial for extended-area mapping where repeated visits to the same regions are common across agents.

3.3.2. Plane-Guided Gaussian Initialization

For an incoming keyframe k (from any agent), we sample pixels on a stride s and cast rays from the camera center $\mathbf{C}$ in the global frame. Using intrinsics (f_x, f_y, c_x, c_y), a pixel (u, v) defines

$$\mathbf{r}_c(u, v) = \begin{bmatrix} (u - c_x)/f_x \\ (v - c_y)/f_y \\ 1 \end{bmatrix}, \qquad \mathbf{r}_w(u, v) = \mathbf{R}_{wc} \frac{\mathbf{r}_c(u, v)}{\|\mathbf{r}_c(u, v)\|}, \tag{10}$$

where $(\mathbf{R}_{wc}, \mathbf{C})$ is the keyframe pose in the global frame (obtained by composing the agent SLAM pose with its Sim(3) alignment). We compute ray–plane intersection by

$$\lambda = \frac{d - \mathbf{n}^\top \mathbf{C}}{\mathbf{n}^\top \mathbf{r}_w}, \qquad \mathbf{x} = \mathbf{C} + \lambda \mathbf{r}_w. \tag{11}$$

We reject unstable cases (near-parallel rays, $\lambda \leq 0$, overly large λ) and enforce a plane-consistency check $\left|\mathbf{n}^\top \mathbf{x} - d\right| \leq \tau_\pi$. Each accepted point is mapped to its grid cell (i, j); if $(i, j) \in \mathcal{C}_{\text{covered}}$ we skip it, otherwise we insert a new Gaussian at $\mathbf{x}$ with the sampled color and mark the cell as covered. This yields a coverage-driven densification policy that is both simple and scalable.

3.3.3. Gradient-Aware Adaptive Densification

Coverage alone is insufficient, as geometrically inaccurate or photometrically inconsistent regions may still exist after initial densification. In our system, such non-converged regions are indicated by persistent position gradients during Gaussian optimization.

Let $\mathcal{G}(c)$ denote the set of Gaussians whose means lie in grid cell c. For each Gaussian g_j with mean $\boldsymbol{\mu}_j$, differentiable splatting yields gradients $\nabla_{\boldsymbol{\mu}_j} L$. We aggregate them into a cell-level indicator:

$$\Gamma(c) \;=\; \mathrm{EMA}\left(\frac{1}{|\mathcal{G}(c)|}\sum_{g_j\in\mathcal{G}(c)}\phi\Big(\left\|\nabla_{\boldsymbol{\mu}_j}L\right\|_2\Big)\right), \tag{12}$$

where $\phi(\cdot)$ is a robust reduction function and $\mathrm{EMA}(\cdot)$ stabilizes the signal across iterations.

If $\Gamma(c) > \tau_\Gamma$, the cell is considered under-converged, and additional plane-guided densification is triggered within c, even if it has been previously covered. This strategy increases local representational capacity in difficult regions while avoiding unnecessary updates in already stable areas.

The overall procedure is summarized in Algorithm 1. Overall, this mechanism enables targeted refinement in under-constrained regions and improves the stability of incremental orthomosaic generation over extended areas.

Algorithm 1 Plane-guided grid-based collaborative densification.

Require: Gaussian map $\mathcal{M}$, incoming keyframe $\mathbf{I}_k$ and its global pose, ground plane Π, grid size g, covered-cell set $\mathcal{C}_{\text{covered}}$, and gradient threshold τ_Γ
Ensure: Updated Gaussian map $\mathcal{M}$
Update the gradient indicator $\Gamma(c)$ for each occupied grid cell c
for each sampled pixel (u, v) in $\mathbf{I}_k$ **do**
  Project (u, v) onto the ground plane Π to obtain the 3D point $\mathbf{x}$
  if $\mathbf{x}$ is invalid **then**
    continue
  end if
  Transform $\mathbf{x}$ into the plane-local coordinate $\mathbf{q}$
  Determine its grid cell:
  $c \leftarrow \left(\left\lfloor \frac{q_x}{g} \right\rfloor, \left\lfloor \frac{q_y}{g} \right\rfloor\right)$
  if $c \notin \mathcal{C}_{\text{covered}}$ or $\Gamma(c) > \tau_\Gamma$ **then**
    Initialize a Gaussian at $\mathbf{x}$ using color $\mathbf{I}_k(u, v)$
    Add the new Gaussian to $\mathcal{M}$
    $\mathcal{C}_{\text{covered}} \leftarrow \mathcal{C}_{\text{covered}} \cup \{c\}$
  end if
end for
return the updated Gaussian map $\mathcal{M}$

3.4. Visibility-Aware Gaussian Pruning

As multi-UAV mapping scales to large environments, redundant Gaussian primitives can accumulate rapidly due to repeated observations and long-term incremental updates. To control model growth without damaging map consistency, we propose a visibility-aware Gaussian pruning strategy that removes low-contribution Gaussians while preserving primitives supported by the SLAM co-visibility structure.

3.4.1. Co-Visibility-Aware Candidate Selection

Let $\mathcal{W}$ be a recent keyframe window. To prevent the pruning decision from being overly myopic, we augment $\mathcal{W}$ using the SLAM co-visibility graph $\mathcal{G} = (\mathcal{K}, \mathcal{E})$, where an edge (k, k') indicates that keyframes k and k' share sufficient co-visible map points. For each $k \in \mathcal{W}$, we sort its co-visible keyframes according to the co-visibility weight and select the top-M keyframes to form $\mathcal{N}_{\text{cov}}(k)$. The augmented keyframe set is then defined as follows:

$$\widetilde{\mathcal{W}} \;=\; \mathcal{W} \cup \bigcup_{k\in\mathcal{W}} \mathcal{N}_{\text{cov}}(k), \tag{13}$$

which expands supervision to structurally related views, such as revisited regions and overlapping flight strips, without requiring global traversal.

For each keyframe $k \in \widetilde{\mathcal{W}}$, the rasterizer provides the indices of visible Gaussians:

$$\mathcal{V}_k = \{ j \mid r_{j,k} > 0 \}, \tag{14}$$

where $r_{j,k}$ is the rendered 2D radius of Gaussian g_j in view k. Gaussians with $r_{j,k} = 0$ are outside the view frustum or have no effective image footprint. We then form the union visibility set:

$$\mathcal{V} = \bigcup_{k \in \widetilde{\mathcal{W}}} \mathcal{V}_k. \tag{15}$$

At each pruning step, pruning is restricted to $\mathcal{V}$, which avoids removing Gaussians that are temporarily unobserved by the current local window. Meanwhile, the co-visibility augmentation ensures that primitives supported by nearby SLAM-connected views are also evaluated.

3.4.2. Gradient-Based Contribution Scoring and Pruning

To estimate the importance of each Gaussian in the current optimization process, we adopt a lightweight gradient-based sensitivity measure derived from the differentiable rendering pipeline. Instead of introducing additional optimization objectives, we reuse first-order gradients computed during the standard backward pass.

For each keyframe k, let L_k denote the photometric reconstruction loss. We introduce an auxiliary scalar contribution weight w_j for each Gaussian g_j and evaluate how sensitively the loss responds to perturbations of this weight:

$$s_{j,k} = \left| \frac{\partial L_k}{\partial w_j} \right|, \qquad S_j = \sum_{k \in \widetilde{\mathcal{W}}} s_{j,k}. \tag{16}$$

The accumulated score S_j measures the rendering influence of Gaussian g_j across recent and co-visible views. Within the candidate set $\mathcal{V}$, a small S_j indicates that the Gaussian has limited influence on the reconstruction objective and can be assigned higher pruning priority.

Given a pruning ratio $\rho \in [0, 1]$, we compute a threshold τ as the ρ-quantile of $\{S_j \mid j \in \mathcal{V}\}$, implemented efficiently via `kthvalue`. The pruning mask is defined as

$$m_j = \mathbb{I}(j \in \mathcal{V}) \wedge \mathbb{I}(S_j \leq \tau), \tag{17}$$

and Gaussians with $m_j = 1$ are removed, with their optimizer states updated accordingly. If $\mathcal{V}$ is empty, pruning is skipped. This adaptive thresholding avoids manually tuning absolute criteria and remains robust across varying scene scales and observation densities. It should be noted that the sparse SLAM map and the Gaussian representation are maintained independently. Sparse SLAM map points are used to initialize Gaussian primitives, whereas Gaussian pruning only removes primitives from the rendering representation and their associated optimizer states. It does not modify or remove the underlying SLAM map points.

The overall procedure is summarized in Algorithm 2. Overall, this strategy combines rendering visibility, gradient-based sensitivity, and SLAM co-visibility support. Pruning is therefore guided by both geometric observability and optimization relevance, enabling effective redundancy suppression while preserving structurally important map content.

Algorithm 2 Visibility-aware adaptive Gaussian pruning.

Require: Gaussian primitives $\{g_j\}_{j=1}^{N}$, recent keyframe window $\mathcal{W}$, co-visibility graph $\mathcal{G} = (\mathcal{K}, \mathcal{E})$, number of neighbors M, and pruning ratio ρ
Ensure: Pruned Gaussian map
Retrieve the top-M co-visible neighbors $\mathcal{N}_{\text{cov}}(k)$ for each $k \in \mathcal{W}$
Construct the augmented keyframe set:
$\widetilde{\mathcal{W}} \leftarrow \mathcal{W} \cup \bigcup_{k \in \mathcal{W}} \mathcal{N}_{\text{cov}}(k)$
Initialize $\mathcal{V} \leftarrow \varnothing$ and $S_j \leftarrow 0$ for all j
for each keyframe $k \in \widetilde{\mathcal{W}}$ **do**
  Render the Gaussian map and obtain the visible set $\mathcal{V}_k$
  $\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{V}_k$
  for each $j \in \mathcal{V}_k$ **do**
    Compute the rendering contribution $s_{j,k}$ according to the contribution-gradient formulation
    $S_j \leftarrow S_j + s_{j,k}$
  end for
end for
if $\mathcal{V} = \varnothing$ **then**
  return without pruning
end if
Compute the adaptive threshold:
$\tau \leftarrow \text{Quantile}_{\rho}(\{S_j \mid j \in \mathcal{V}\})$
Remove Gaussians satisfying $j \in \mathcal{V}$ and $S_j \leq \tau$
Update the corresponding optimizer states
return the pruned Gaussian map

3.5. Orthographic Gaussian Splatting for Orthophoto Rendering

To export the optimized global Gaussian map into a metric orthophoto, we adopt an orthographic Gaussian Splatting formulation following Tortho-Gaussian [19]. This module serves as the rendering backend of our system, while the preceding modules ensure that all Gaussian primitives are maintained in a unified GNSS-aligned frame.

Unlike vanilla 3D Gaussian Splatting, which uses perspective projection and introduces depth-dependent scale variation, orthographic projection preserves ground-scale consistency and is therefore suitable for orthophoto generation. Each Gaussian mean is projected to the orthographic image plane using the following:

$$P_o = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{z_f - z_n} & -\frac{z_f + z_n}{z_f - z_n} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{18}$$

where (l, r, b, t) denote the orthographic viewing bounds and (z_n, z_f) are the near–far clipping planes. The bounds are determined by the desired ground sampling distance (GSD) and the scene extents $(X_{\min}, X_{\max}, Y_{\min}, Y_{\max})$, yielding

$$W = \frac{X_{\max} - X_{\min}}{\text{GSD}_x}, \quad H = \frac{Y_{\max} - Y_{\min}}{\text{GSD}_y}. \tag{19}$$

This establishes a metric mapping between the global world frame and orthophoto pixels.

As orthographic projection is affine, the 3D covariance Σ of each Gaussian is mapped to the image plane using a constant Jacobian:

$$J_o = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{20}$$

The projected covariance defines the elliptical footprint of each Gaussian on the orthographic image plane. With the projected means and covariances, all Gaussians are splatted onto the orthophoto plane and accumulated using standard front-to-back α-blending. Because all primitives are optimized in a unified GNSS-aligned coordinate frame, the final orthographic rendering directly produces a globally consistent orthophoto without explicit image stitching or post hoc blending.

4. Results

4.1. Experimental Setup

4.1.1. Datasets and Multi-Agent Evaluation Protocol

We evaluate the proposed method on two aerial datasets, including the public DroneMap-NPU dataset [10] and a self-collected UAV dataset, covering diverse environments and scales. In practical multi-UAV aerial mapping, the survey area is typically divided into adjacent lawn-mower flight strips, with different strips assigned to different UAVs. Accordingly, for controlled multi-agent evaluation, each continuous sequence is divided according to its flight strips, and each strip is treated as an independently initialized agent, as illustrated in Figure 5. Overlapping flight trajectories are not required by our system; however, overlapping visual observations enable inter-agent visual loop closures to refine the relative transformations.

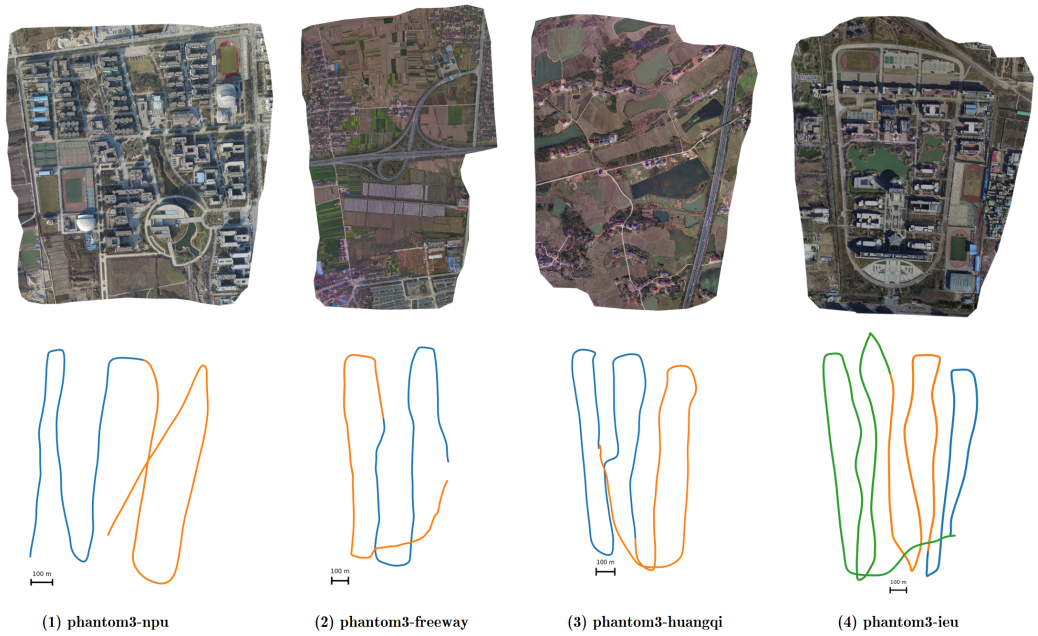


Figure 5. Illustration of the construction of multi-agent orthophoto mapping scenarios from real UAV sequences. Each trajectory is partitioned into several overlapping flight strips, where different colors denote different agents. The corresponding orthomosaics are shown above, and the partitioned trajectories are visualized below.

We adopt the DroneMap-NPU dataset, which was captured using a customized hexacopter platform equipped with a DJI Phantom 3 camera (SZ DJI Technology Co., Ltd.,

Shenzhen, China). The dataset covers a wide range of real-world aerial scenarios, including mountainous areas, rivers, highways, residential regions, and grasslands, with significant variations in terrain and elevation. Each image is associated with GNSS measurements, providing global positioning information for geo-referenced mapping. The DroneMap-NPU dataset provides temporally ordered image frames extracted from the original UAV videos. To balance mapping accuracy and computational efficiency, approximately one out of every four frames is uniformly selected as input. The self-collected videos are recorded at 30 fps, and all frames are sequentially decoded for processing.

To further evaluate the performance of our system on real UAV imagery, we collect additional aerial video sequences using a DJI Mini 5 Pro platform. The data are captured as 1080p video sequences with synchronized onboard GNSS measurements, from which image frames are extracted for processing. The dataset covers several aerial mapping scenarios with overlapping flight regions and diverse structures. All images are associated with GNSS data, enabling geo-consistent mapping and evaluation. Both the public and self-collected datasets provide consumer-grade onboard GNSS measurements without RTK corrections. Detailed statistics are reported in Table 1. In this Table, H-Max denotes the maximum UAV flight altitude recorded during data acquisition. As the UAV follows a planned lawn-mower trajectory and maintains an approximately constant altitude after takeoff, H-Max is representative of the flight altitude over most of the mapping sequence. As the original datasets do not directly provide GSD values, the theoretical GSD reported in Table 1 is calculated from H-Max and the calibrated camera focal lengths as

$$\mathrm{GSD} = \frac{1}{2}\left(\frac{\text{H-Max}}{f_x} + \frac{\text{H-Max}}{f_y}\right), \tag{21}$$

where f_x and f_y denote the horizontal and vertical focal lengths in pixels, respectively.

Table 1. Details of our experimental datasets. "NPU" refers to the NPU-DroneMap dataset, and "Self" refers to our self-collected datasets. H-Max denotes the maximum UAV flight altitude recorded during data acquisition, and GSD denotes the theoretical ground sampling distance calculated at H-Max.

Dataset	Sequence	H-Max (m)	GSD (cm/pixel)	Area (km^2)	Image Num	Resolution	Segments
NPU-DroneMap	phantom3-huangqi	222.30	18.22	1.313	14,776	1080P	2
	phantom3-freeway	258.30	21.17	1.457	17,120	1080P	2
	phantom3-ieu	282.30	23.14	1.524	21,969	1080P	3
	phantom3-npu	254.50	20.86	1.598	13,983	1080P	2
Self-collect	railway	220.00	7.80	0.303	3193	1080P	3
	farmland	120.00	4.26	0.503	4193	1080P	2
	residential	210.00	7.45	0.416	2971	1080P	3

4.1.2. Implementation Details

Our system is implemented in C++ and CUDA, and runs on a desktop platform equipped with an Intel i7-13900K CPU, 32 GB RAM, and an NVIDIA RTX 3090 GPU with 24 GB memory. For plane-guided collaborative densification, the grid resolution is set to $g = 5$ m. For gradient-aware re-densification, the threshold is set to $\tau_\Gamma = 0.01$. For visibility-aware pruning, the pruning ratio is fixed to $\rho = 0.35$, and the co-visibility expansion considers the top-$M = 10$ neighboring keyframes. During multi-UAV coordination, the geographic gating radius is set to $r_{\mathrm{geo}} = 20$ m. The orthophoto renderer is triggered at 10 Hz for incremental map updates. These parameters are empirically selected based on the trade-off between reconstruction quality, computational efficiency, and memory consumption. Specifically, the grid resolution g determines the spatial granularity of

plane-guided Gaussian initialization. The pruning ratio ρ controls the balance between redundant Gaussian removal and scene structure preservation, while the geographic gating radius r_{geo} is selected according to GNSS positioning uncertainty and inter-agent overlap to ensure reliable candidate retrieval. The same parameter settings are maintained across all experiments for fair comparison. In the current evaluation, the multi-agent data streams are replayed on the ground workstation to provide a controlled and repeatable experimental setting. Therefore, actual wireless communication latency is not modeled.

4.1.3. Comparison Methods

We compare the proposed method with three groups of representative baselines. First, commercial photogrammetry software, including Agisoft Metashape (version 2.0) [7], ContextCapture (version 23.00.00.1317) [8], and Pix4Dmapper (version 4.8) [9], is used as strong offline baselines for high-quality orthophoto generation. Second, SLAM-based image mosaicking methods, including Map2DFusion [10] and EfficientOrtho [13], are selected to evaluate online incremental mapping performance. Third, representative 3DGS-SLAM methods, including Photo-SLAM [16] and CartGS [15], are included to assess the effectiveness of Gaussian-based scene representation. To the best of our knowledge, no publicly available multi-UAV orthophoto mapping system is available for direct comparison. Therefore, these comparison methods are evaluated in their original single-agent mode whenever applicable. For 3DGS-SLAM baselines, we apply the same orthographic rendering procedure to their reconstructed Gaussian maps to generate orthophoto results for comparison.

4.2. Geo-Consistency Evaluation

We evaluate the geo-consistency of the orthophotos generated by the proposed method using both public datasets with ground control points (GCPs) and simulated multi-UAV settings.

We adopt the phantom3-npu sequence from the DroneMap-NPU dataset, which provides six GCPs for quantitative evaluation. The spatial distribution and detailed information of these GCPs are illustrated in Figure 6. as the dataset is originally captured by a single UAV, we simulate a multi-UAV scenario by splitting the sequence into two sub-trajectories with overlapping regions.

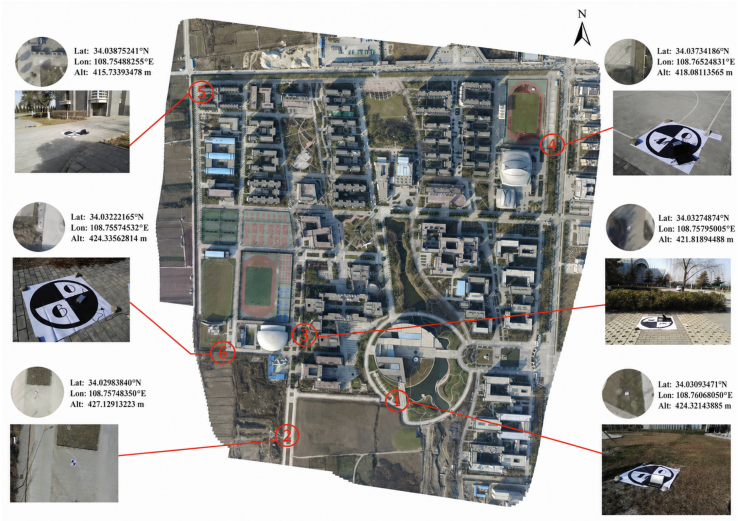


Figure 6. Illustration of ground control points (GCPs) in the phantom3-npu sequence. The locations of GCPs are marked on the orthomosaic, with corresponding close-up views and geographic coordinates shown for each point. The coordinates are represented by latitude (°N), longitude (°E), and altitude (m), and the north arrow indicates the geographic orientation.

We compare the proposed method with representative photogrammetry software, including Agisoft Metashape, ContextCapture, and Pix4Dmapper, which serve as strong offline baselines for geo-referenced orthomosaic generation. In addition, we include a recent online incremental mapping method, EfficientOrtho, for comparison. As these methods only support single-UAV mapping, all baseline methods are evaluated using the original (non-split) sequence.

The geo-consistency is evaluated by measuring the absolute planimetric error at GCP locations on the generated orthomosaic. For each GCP, we compute the Euclidean distance between the predicted position $(X_i^{\text{pred}}, Y_i^{\text{pred}})$ and the reference position $(X_i^{\text{gt}}, Y_i^{\text{gt}})$:

$$e_i = \sqrt{(X_i^{\text{pred}} - X_i^{\text{gt}})^2 + (Y_i^{\text{pred}} - Y_i^{\text{gt}})^2}. \tag{22}$$

The predicted GCP coordinates are obtained by manually locating the corresponding control points on the generated geo-referenced orthomosaic using QGIS, while the reference coordinates are obtained from the GCP annotations provided by the NPU-DroneMap dataset. The overall accuracy is then summarized using the root mean square error (RMSE) over all GCPs:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2}, \tag{23}$$

where $N = 6$ is the number of GCPs.

The quantitative results are reported in Table 2. Compared with offline photogrammetry methods, including Metashape, ContextCapture, and Pix4Dmapper, the proposed method achieves competitive geo-consistency while significantly reducing processing time. Specifically, although Pix4Dmapper attains the best RMSE of 1.51 m, our method achieves an RMSE of 2.32 m with a much shorter runtime. This demonstrates that the proposed method provides a favorable trade-off between accuracy and efficiency, avoiding the heavy computational cost of offline pipelines while maintaining reliable geo-consistency.

Table 2. Comparison of GCP errors (m), RMSE (m), and total processing time (minute) among different methods.

GCP	ContextCapture	MetaShape	Pix4Dmapper	EfficientOrtho	Ours
1	2.76	2.18	1.26	3.35	1.92
2	3.53	2.87	1.74	3.62	2.63
3	3.08	2.44	1.56	3.41	2.31
4	3.34	2.71	1.39	3.48	2.17
5	2.83	2.29	1.41	3.29	2.35
6	3.29	2.61	1.62	3.55	2.48
RMSE	3.15	2.53	1.51	3.45	2.32
Time	43.2	35.5	37.2	4.6	3.9

Furthermore, compared with the SLAM-based baseline EfficientOrtho, the proposed method achieves both higher accuracy and better efficiency. EfficientOrtho produces an RMSE of 3.45 m with a runtime of 4.6 min, while our method reduces the RMSE to 2.32 m and further shortens the runtime to 3.9 min. These results demonstrate that the proposed method effectively improves both accuracy and efficiency over existing incremental mapping approaches. By leveraging multi-UAV collaboration, the system achieves faster processing while maintaining strong geo-consistency, highlighting its advantage for online orthomosaic generation in aerial mapping scenarios.

4.3. Orthomosaic Quality Evaluation

A high-quality orthomosaic, in addition to geo-consistency, should also satisfy two essential requirements: high visual fidelity and strong geometric consistency. Therefore, we evaluate the proposed method from both photometric and geometric perspectives.

For photometric quality, we adopt two widely used image similarity metrics in 3DGS, namely, PSNR and SSIM. As orthomosaics lack ground truth, we evaluate reconstruction quality using rendered images from keyframe poses. As both rendering and orthomosaic generation rely on the same 3D Gaussian representation, PSNR and SSIM can serve as effective indirect indicators of orthomosaic visual quality.

Given a reference image I and a rendered image $\hat{I}$ with N pixels, the mean squared error (MSE) is defined as

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(I_i - \hat{I}_i)^2. \tag{24}$$

The PSNR is then computed as

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{L^2}{\mathrm{MSE}}\right), \tag{25}$$

where L denotes the maximum possible pixel value. The SSIM between I and $\hat{I}$ is defined as

$$\mathrm{SSIM}(I,\hat{I}) = \frac{(2\mu_I\mu_{\hat{I}} + C_1)(2\sigma_{I\hat{I}} + C_2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + C_1)(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2)}, \tag{26}$$

where μ_I and $\mu_{\hat{I}}$ are the mean intensities, σ_I^2 and $\sigma_{\hat{I}}^2$ are the variances, and $\sigma_{I\hat{I}}$ is the covariance. C_1 and C_2 are constants for numerical stability.

To further demonstrate the visual effectiveness of the proposed method, we compare it with two representative 3DGS-SLAM approaches, namely, PhotoSLAM [16] and CartGS [15]. As these methods are not specifically designed for orthomosaic generation, we apply an orthographic projection to their reconstructed Gaussian representations to generate orthomosaic results for fair comparison. As shown in Table 3, the proposed method consistently outperforms both PhotoSLAM and CartGS across all sequences in terms of PSNR and SSIM. For example, on the phantom3-npu sequence, our single-UAV setting achieves a PSNR of 29.108 dB, compared with 25.925 dB and 26.625 dB for PhotoSLAM and CartGS, respectively. Similar improvements can be observed on all other sequences, demonstrating the superior visual reconstruction capability of the proposed method.

Table 3. Quantitative comparison in terms of SSIM and PSNR on different datasets. Bold values indicate the best performance.

Method	Npu		Ieu		Freeway		Railway		Farmland		Residential	
	SSIM	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**
PhotoSLAM	0.742	25.925	0.776	25.375	0.761	28.972	0.753	26.431	0.767	27.891	0.749	26.326
CartGS	0.798	26.625	0.798	27.131	0.818	30.731	0.805	28.314	0.812	29.648	0.794	27.137
Ours (Single)	**0.837**	**29.108**	**0.861**	**30.443**	**0.884**	**33.012**	**0.872**	**32.179**	**0.866**	**31.251**	**0.845**	**29.072**
Ours (Multi)	0.826	28.652	0.853	30.332	0.874	31.891	0.861	31.681	0.855	30.412	0.832	28.767

In addition, the single-UAV configuration of our method achieves the best performance among all compared approaches. This is mainly because the optimization process can focus on a single trajectory, allowing more sufficient Gaussian refinement and leading to higher reconstruction quality. For the multi-UAV setting, although the PSNR and SSIM values are slightly lower than those of the single-UAV configuration, the performance still consistently

exceeds that of the compared methods. The slight degradation is mainly due to the limited optimization time per agent and the increased complexity introduced by multi-UAV fusion. Nevertheless, the multi-UAV configuration maintains strong visual quality while enabling significantly improved efficiency and coverage, as demonstrated in the previous section.

In addition to photometric quality, we further evaluate relative structural consistency using a line-based metric, which measures the preservation of structural shapes and spatial layout in the generated orthomosaic. As the datasets used in this work do not provide reliable geometric ground truth, we use the orthomosaics generated by representative photogrammetry software, namely, Metashape and ContextCapture, as reference baselines rather than ground truth. Specifically, we manually select eight pairs of corresponding line segments from the phantom3-npu and phantom3-ieu sequences, covering representative structural regions such as roads, building edges, and field boundaries. These selected line segments are visualized in Figure 7. The endpoints of each line segment are manually identified on the geo-referenced orthomosaics using QGIS, and the corresponding segment lengths are calculated based on the projected coordinate system of the orthomosaics. For each pair, we compute the ratio of their lengths to evaluate geometric consistency. The absolute error (AE) and relative error (RE) are defined as

$$\mathrm{AE} = |R_{\mathrm{pred}} - R_{\mathrm{ref}}|, \quad \mathrm{RE} = \frac{|R_{\mathrm{pred}} - R_{\mathrm{ref}}|}{R_{\mathrm{ref}}} \times 100\%, \tag{27}$$

where R_{pred} and R_{ref} denote the predicted and reference length ratios, respectively.

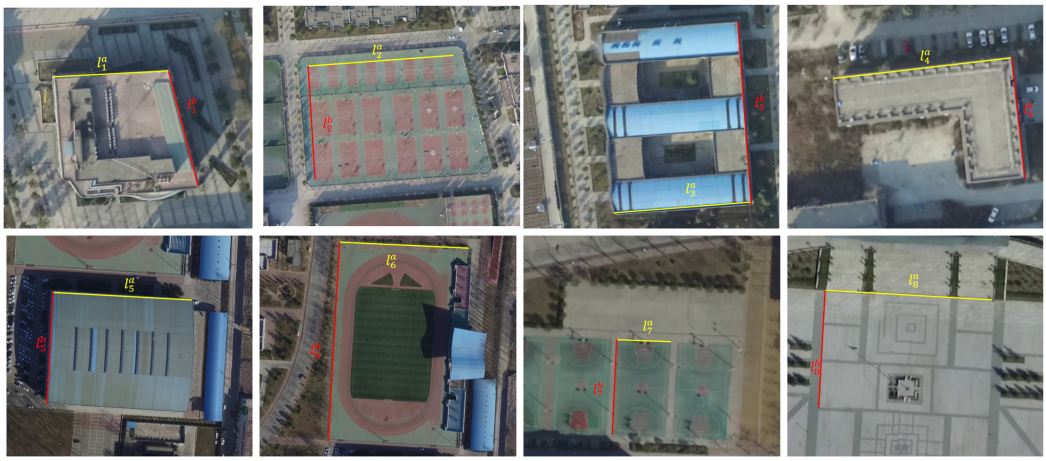


Figure 7. Selected line segment pairs for geometric consistency evaluation on the phantom3-npu and phantom3-ieu sequences.

The results are reported in Table 4. It can be observed that the proposed method achieves consistently low differences across the selected line segment pairs. These results indicate that the proposed method preserves relative structural proportions consistent with those generated by the reference photogrammetry pipelines. In particular, most relative errors remain below 1%, demonstrating accurate preservation of spatial relationships and scale consistency. The relatively larger errors observed in Line segments 4 and 5 are mainly attributed to their locations near the boundary regions of the UAV flight trajectory, where fewer observations and weaker viewpoint constraints are available.

Overall, the proposed method preserves relative structural proportions consistent with those generated by the reference photogrammetry pipelines while maintaining an efficient incremental mapping process. Table 4 provides a complementary relative comparison rather than an evaluation of absolute geometric accuracy.

Table 4. Relative structural consistency comparison with Metashape and ContextCapture. The ID denotes the index of the line segment pair (l_i^a, l_i^b) illustrated in Figure 7.

ID	Ours	MetaShape			ContextCapture		
	Ratio	Ratio	Rel. Err. (%)	Abs. Err.	Ratio	Rel. Err. (%)	Abs. Err.
1	0.95192	0.95796	0.631	0.00604	0.95681	0.511	0.00489
2	1.30147	1.29248	0.696	0.00899	1.29513	0.490	0.00634
3	0.76521	0.76281	0.315	0.00240	0.76410	0.145	0.00111
4	1.51683	1.53261	1.030	0.01578	1.53620	1.261	0.01937
5	1.22209	1.22860	0.530	0.00651	1.23090	0.716	0.00881
6	0.62748	0.62510	0.381	0.00238	0.62600	0.236	0.00148
7	0.54572	0.54240	0.612	0.00332	0.54420	0.279	0.00152
8	1.40963	1.41092	0.091	0.00129	1.41290	0.231	0.00327
Mean	-	-	0.536	0.00584	-	0.484	0.00585

4.4. Qualitative Evaluation of Orthomosaic Generation

A high-quality orthomosaic should exhibit clear structural boundaries, consistent appearance, and minimal geometric distortion. To qualitatively evaluate the orthomosaic generation performance of the proposed method, we visualize the generated results on multiple sequences from both the NPU-DroneMap dataset and our self-collected datasets.

The qualitative results are shown in Figure 8. It can be observed that the proposed method produces high-quality orthomosaics across diverse scenes, including urban areas, river regions, and farmland environments. In particular, building boundaries are well preserved, texture details remain sharp, and no obvious stitching artifacts or geometric distortions are observed. Furthermore, the generated orthomosaics exhibit strong geometric consistency and visual coherence over aerial mapping scenarios, even in regions with complex structures or viewpoint variations. These results demonstrate the robustness of the proposed method and its capability to produce reliable and high-quality orthomosaic maps in various real-world scenarios.

To further analyze the visual quality in detail, we present zoom-in comparisons of representative regions in Figure 9, including results from Map2DFusion [10], EfficientOrtho [36], Metashape [7], ContextCapture [8], and Pix4Dmapper [9] on the phantom3-npu and phantom3-ieu sequences.

The proposed method produces sharper and more consistent building structures, with clear and well-aligned boundaries. In particular, our method effectively removes facade artifacts and preserves true top-down geometry, resulting in a genuine orthomosaic representation. In contrast, as highlighted by the dashed boxes, other methods suffer from noticeable artifacts caused by limited viewpoints and projection inconsistencies. For example, SLAM-based methods such as Map2DFusion and EfficientOrtho often exhibit blurred textures and structural distortions, while photogrammetry-based methods may introduce facade effects or ghosting artifacts around building edges. These issues lead to inconsistencies in building geometry and degrade the overall visual quality. Our proposed method demonstrates superior capability in preserving structural details and generating artifact-free orthomosaics, especially in challenging regions with complex building layouts.

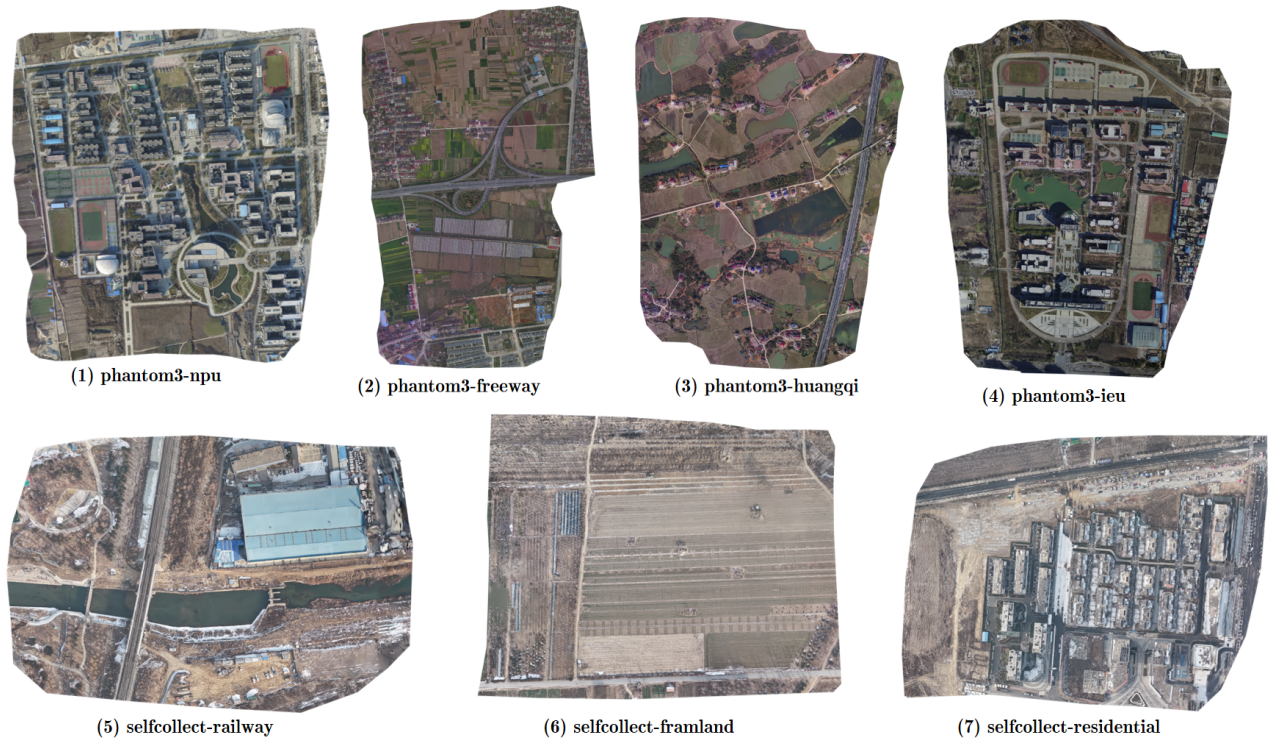


Figure 8. Qualitative orthomosaic results on multiple sequences: (**1**) npu, (**2**) freeway, (**3**) huangqi, (**4**) ieu, (**5**) railway, (**6**) farmland, and (**7**) residential.



Figure 9. Zoom-in comparison of orthomosaic details on the npu and ieu sequences. The proposed method produces clear boundaries without facade artifacts, while other methods exhibit facade effects, ghosting, or structural distortions, as highlighted by dashed boxes.

To further validate the improvements of the proposed method over existing 3DGS-SLAM approaches, we compare it with two representative methods, namely, Photo-SLAM and CartGS. As these methods are not specifically designed for orthomosaic generation [19], we integrate an orthographic rendering module to generate orthomosaics from their reconstructed Gaussian representations for fair comparison.

The qualitative results are shown in Figure 10. It can be observed that Photo-SLAM suffers from severe floating artifacts over large regions on the phantom3-huangqi and phantom3-freeway sequences. These artifacts significantly degrade the visual quality of the generated orthomosaics and lead to inconsistent textures and distorted structures. As highlighted by the dashed boxes, both Photo-SLAM and CartGS produce blurred details and noticeable reconstruction artifacts in local regions. These issues are mainly caused by unstable Gaussian optimization and insufficient geometric constraints, which result in inaccurate modeling of scene structures. In addition, these methods are limited by memory constraints and are unable to reconstruct complete orthomosaics for aerial mapping scenarios. In contrast, the proposed method effectively suppresses floating artifacts and preserves sharper and more consistent structures. The generated orthomosaics exhibit clearer textures, well-defined boundaries, and improved structural integrity across all evaluated scenes.

Overall, these results demonstrate that the proposed method significantly improves the robustness and visual quality of orthomosaic generation within the 3DGS-SLAM framework, especially in challenging aerial mapping environments.

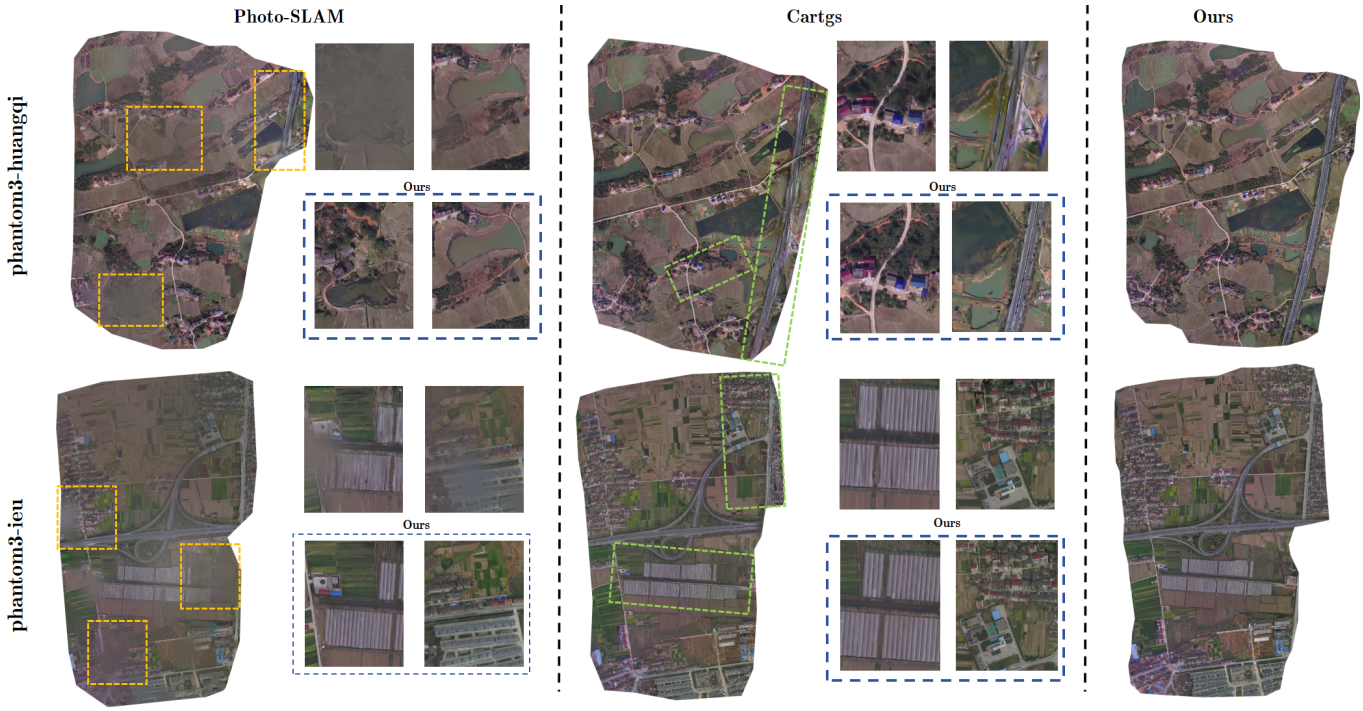


Figure 10. Qualitative comparison of orthomosaic generation using different 3DGS-SLAM methods on the huangqi and freeway sequences. For each case, the left shows the full orthomosaic, while the right presents enlarged views of the regions marked by colored boxes for detailed comparison with our method.

4.5. Multi-UAV Performance Analysis

We evaluate the effectiveness of the proposed multi-UAV framework in terms of coverage, efficiency, and orthomosaic quality by comparing single-UAV and multi-UAV configurations on the same scenes. Experiments are conducted on four representative sequences, namely, phantom3-npu, phantom3-ieu, phantom3-freeway, and phantom3-

huangqi, under both single-UAV and multi-UAV collaborative settings. Coverage is defined as the ratio of processed frames within a fixed time interval to the total number of frames, reflecting mapping progress over time, while efficiency is measured by the total time required to complete orthomosaic generation for the entire scene.

As reported in Table 5, the multi-UAV configuration consistently improves both coverage and efficiency. Specifically, the multi-UAV configuration improves the coverage by approximately 40–60% across all evaluated sequences. Meanwhile, the total mapping time is reduced by about 35–55% (e.g., 6.2 → 4.2 min on *phantom3-npu*), demonstrating the advantage of collaborative mapping for aerial mapping scenarios. The speedup is sub-linear due to the additional synchronization and joint optimization overhead introduced by centralized multi-agent fusion. Despite this, the multi-UAV setting maintains comparable orthomosaic quality, with only a minor PSNR decrease of 0.3–1.1 dB.

Table 5. Comparison of single-UAV and multi-UAV performance on different sequences.

Sequence	Setting	Coverage (%)	Time (min)	PSNR (dB)
phantom3-npu	Single	15.6	6.2	29.13
	Multi	25.6	4.2	28.65
phantom3-ieu	Single	11.9	8.3	30.423
	Multi	18.2	5.5	30.332
phantom3-freeway	Single	17.5	5.7	33.053
	Multi	25.6	3.9	31.891
phantom3-huangqi	Single	16.1	6.2	29.734
	Multi	24.3	4.1	29.491
railway	Single	17.8	5.6	32.179
	Multi	26.3	3.8	31.681
residential	Single	20.4	4.9	29.072
	Multi	30.3	3.3	28.767

Overall, these results demonstrate that the proposed framework effectively balances efficiency and quality, enabling significantly faster orthomosaic generation while maintaining comparable visual fidelity.

4.6. Efficiency Analysis

We evaluate the efficiency of the proposed method from both overall runtime and per-module latency perspectives.

We first compare the total processing time of the proposed method with representative commercial photogrammetry software, including Agisoft Metashape, ContextCapture, and Pix4Dmapper, on several representative sequences, as shown in Table 6. The reported time covers the complete orthomosaic generation pipeline from raw images to final map output. Compared with offline photogrammetry pipelines, which typically require tens of minutes due to global structure-from-motion and dense reconstruction, the proposed method significantly reduces the total runtime from 30–60 min to only 4–8 min across all sequences, achieving a speedup of approximately 6× to 10×. The multi-UAV mode further improves efficiency compared with the single-UAV setting (e.g., 6.2 → 4.2 min on *phantom3-npu*), demonstrating the advantage of collaborative mapping for efficient coverage of extended areas. As offline photogrammetry pipelines differ from online SLAM methods in processing paradigm and output resolution, this comparison is provided only as an application-level reference rather than a strictly equivalent runtime benchmark.

Table 6. Total processing time comparison (in minutes) on representative sequences.

Sequence	Ours (Single)	Ours (Multi)	MetaShape	ContextCapture	Pix4D
npu	6.2	4.2	34.6	54.1	36.1
ieu	8.5	5.5	42.1	64.3	47.7
freeway	5.6	3.9	42.7	59.1	42.1
huangqi	6.2	4.1	39.6	49.5	41.1

For efficiency comparison with online Gaussian-based SLAM methods, we further compare the proposed method with Photo-SLAM and CartGS under identical hardware and input settings, as shown in Table 7. Across all evaluated sequences, the proposed method consistently achieves the highest PSNR while requiring less peak GPU memory, fewer Gaussian primitives, and shorter total runtime. Compared with CartGS, our method improves PSNR by approximately 0.95–1.97 dB, reduces peak memory by 2.6–5.4 GB, and shortens the total runtime by 1.4–1.8 min. These results demonstrate that the proposed collaborative densification and pruning strategies effectively improve mapping quality while controlling model redundancy and computational cost.

Table 7. Efficiency and mapping-quality comparison of online Gaussian-based SLAM methods. PSNR is reported in dB, peak memory in GB, the number of Gaussians in millions, and runtime in minutes.

Sequence	Method	PSNR	Peak Memory	Gaussians	Runtime
freeway	Photo-SLAM	28.972	17.6	0.86	5.8
	CartGS	30.731	16.5	0.62	5.4
	Ours (Multi)	31.891	11.1	0.41	3.9
huangqi	Photo-SLAM	27.205	14.6	0.71	6.2
	CartGS	28.541	13.0	0.57	5.9
	Ours (Multi)	29.491	9.6	0.38	4.1
railway	Photo-SLAM	28.593	15.3	0.75	5.6
	CartGS	29.714	12.6	0.52	5.4
	Ours (Multi)	31.681	9.2	0.37	3.8
residential	Photo-SLAM	25.631	14.3	0.67	5.1
	CartGS	27.526	11.2	0.42	4.7
	Ours (Multi)	28.767	8.6	0.31	3.3

We further analyze the runtime of each core module, including tracking, 3D Gaussian optimization, and orthomosaic rendering, as summarized in Table 8. Tracking is executed for each incoming frame with an average latency of 10.4 ± 2.2 ms, enabling real-time pose estimation. The 3D Gaussian optimization runs continuously in the backend with a low per-iteration cost (8.9 ± 1.2 ms), supporting efficient incremental updates of the global representation.

Table 8. Runtime breakdown of the proposed system. Latency is reported as mean ± standard deviation.

Module	Latency	Frequency
Tracking	10.4 ± 2.2 ms	per frame
3DGS Optimization	8.9 ± 1.2 ms	continuous
Orthomosaic Rendering	102.7 ± 10.5 ms	∼10 Hz

Orthomosaic rendering is performed asynchronously with a bounded latency of 102.7 ± 10.5 ms per update (approximately 10 Hz), which is sufficient for online map

visualization without introducing significant delay. This decoupled design allows the system to maintain real-time tracking while continuously updating the global orthomosaic.

Overall, these results demonstrate that the proposed framework achieves high mapping quality, computational efficiency, and scalability.

4.7. Ablation Study

To further analyze the effectiveness of each core component in the proposed framework, we conduct ablation studies on three key modules: (1) GNSS-prior multi-UAV alignment with visual loop refinement, (2) plane-guided collaborative densification, and (3) visibility-aware Gaussian pruning.

Specifically, we construct the following variants:

- Full: The complete proposed method.
- GNSS-only: Replacing the proposed GNSS-prior visual loop refinement with GNSS-based coarse alignment only.
- w/o densification: Removing the plane-guided collaborative densification strategy.
- w/o pruning: Removing the visibility-aware Gaussian pruning mechanism.

We evaluate these variants using PSNR, SSIM, runtime, Gaussian count, and peak GPU memory consumption. Quantitative ablation experiments are conducted on the representative phantom3-ieu sequence, and the results are summarized in Table 9. The computation of PSNR and SSIM follows the same protocol as in the quantitative evaluation.

Table 9. Ablation study of the proposed method. "GNSS-only" removes visual loop refinement, "w/o densification" removes plane-guided collaborative densification, and "w/o pruning" removes visibility-aware adaptive pruning. † indicates that the w/o pruning variant exceeds GPU memory limits (24 GB) and is only partially evaluated on keyframes before overflow. N/A denotes that the result is not available due to incomplete reconstruction. ↑ indicates that higher values are better, ↓ indicates that lower values are better, and # denotes the number of Gaussians.

Method	PSNR ↑	SSIM ↑	Time (min) ↓	# Gaussians ↓	Peak GPU Mem. (GB) ↓
Full (Ours)	**30.332**	**0.853**	5.3	1.64 M	14.3
GNSS-only	27.24	0.794	5.1	1.67 M	15.1
w/o densification	27.86	0.812	**4.8**	**1.41 M**	**14.2**
w/o pruning	30.128 †	0.842 †	N/A	N/A	>24

4.7.1. Impact of GNSS-Prior Visual Loop Refinement

We first analyze the role of inter-agent visual loop refinement by comparing the full method with the GNSS-only variant.

As shown in Table 9, removing visual loop refinement leads to a significant decrease in PSNR (30.332 → 27.24) and SSIM (0.853 → 0.794), indicating substantial degradation in reconstruction quality.

In our unified Gaussian representation, all observations from multiple UAVs are jointly optimized within a single scene. Under GNSS-only initialization, residual inter-agent misalignment propagates into the 3DGS optimization process, resulting in inconsistent Gaussian updates across overlapping regions. This leads to blurred structures, duplicated textures, and ghosting artifacts, as illustrated in Figure 11.

These results demonstrate that while GNSS provides a useful coarse global reference, it is insufficient to ensure accurate multi-UAV consistency. The proposed visual loop refinement effectively reduces residual misalignment and significantly improves the final orthophoto generation quality.

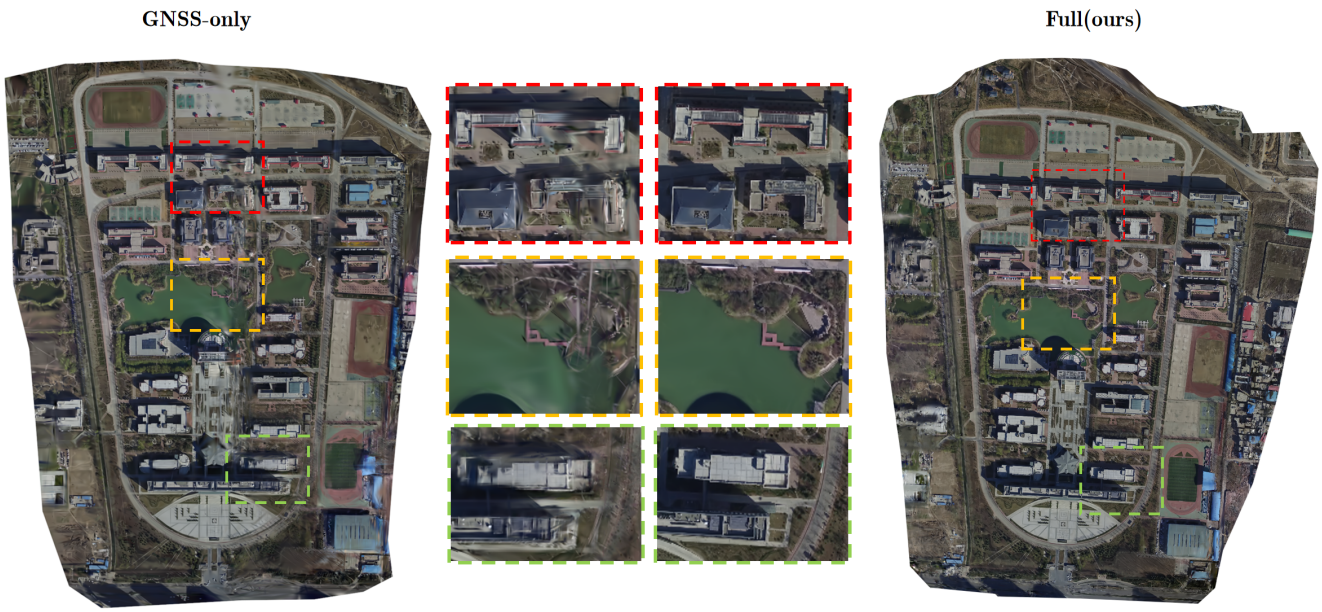


Figure 11. Qualitative comparison of GNSS-only alignment and the full method. (**Left**) Orthomosaic generated using GNSS-only alignment. (**Right**) Result of the full method with inter-agent visual loop refinement. Zoomed-in regions highlight that GNSS-only alignment suffers from misalignment and blurred structures, while the proposed method achieves more accurate alignment and sharper orthomosaic details.

4.7.2. Impact of Plane-Guided Collaborative Densification

We next evaluate the contribution of the proposed plane-guided collaborative densification strategy. As shown in Table 9, removing this module leads to noticeable decreases in PSNR (30.332 → 27.86) and SSIM (0.853 → 0.812), confirming its importance for orthophoto generation quality. Although removing densification reduces runtime, it substantially degrades PSNR and SSIM, indicating that the proposed densification improves mapping quality at an acceptable computational cost.

Qualitative results are illustrated in Figure 12. The left subfigure shows the raw keyframe together with the tracked map points, while the right subfigure presents the instantaneous rendered image at the current optimization stage, reflecting both convergence behavior and robustness in challenging regions. Without densification, the visual quality of the online rendered images is significantly degraded in areas with sparse map points, such as low-texture ground regions and scene boundaries. These regions exhibit blurred textures, incomplete structures, and reduced sharpness. In contrast, the proposed densification strategy produces clearer structures and more consistent details in online, indicating faster convergence and improved robustness in under-constrained regions.

This improvement can be attributed to the fact that conventional densification lacks sufficient geometric support in such regions, leading to slow convergence and under-representation. By leveraging global planar priors and coverage-aware expansion, the proposed method effectively introduces additional Gaussian primitives in poorly constrained areas, thereby enhancing both reconstruction completeness and optimization efficiency.

4.7.3. Impact of Visibility-Aware Adaptive Pruning

Finally, we analyze the effect of the visibility-aware adaptive pruning mechanism. As shown in Table 9, removing this module results in a substantial increase in GPU memory consumption, exceeding the 24 GB limit and preventing full orthophoto generation.

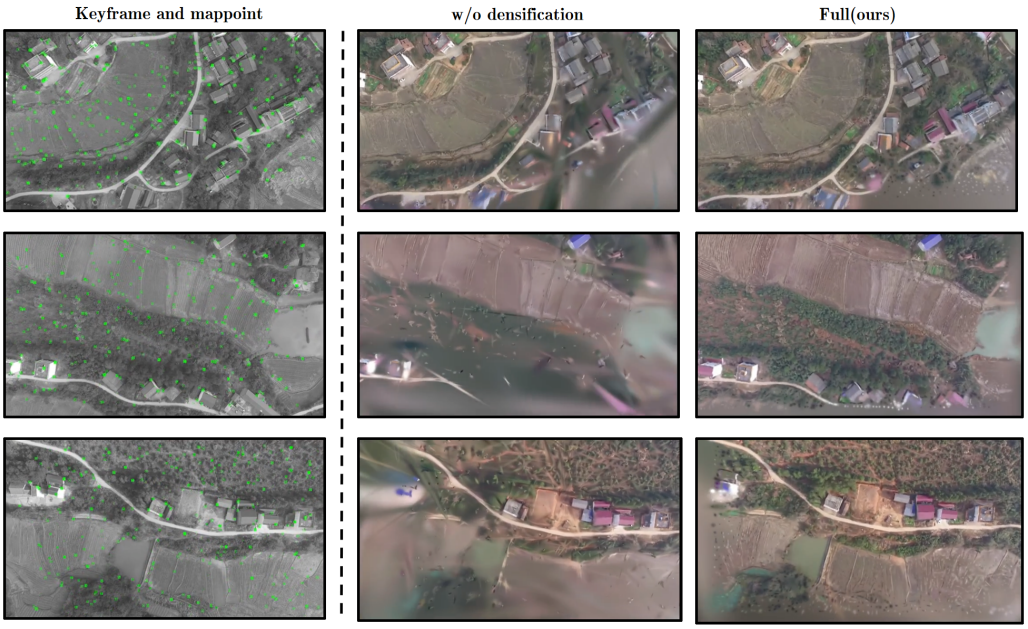


Figure 12. Qualitative comparison of plane-guided collaborative densification. From left to right: The input keyframe with tracked map points, instantaneous rendering without densification, and the result of the full method. Green dots indicate the tracked map points, while the dashed line separates the input observations from the rendering results.

Although the w/o pruning variant achieves comparable PSNR (30.128) and SSIM (0.842) under partial evaluation, it fails to complete the full mapping process due to excessive memory usage. This indicates that the accumulated redundant Gaussian primitives significantly increase computational cost and hinder scalability.

In contrast, the proposed pruning strategy effectively controls the number of Gaussians (1.64 M) and maintains peak GPU memory at a manageable level (14.3 GB), while preserving high orthophoto generation quality. These results highlight that pruning is not only beneficial but is essential for enabling scalable multi-UAV mapping.

Overall, the ablation results confirm that each component plays a complementary role in the proposed framework. The GNSS-prior visual loop refinement ensures global geometric consistency across multiple UAVs, the plane-guided densification improves orthophoto quality in challenging regions, and the visibility-aware pruning maintains scalability and efficiency. Their combination enables robust and online orthomosaic generation.

5. Discussion

The proposed framework addresses the limitations of existing UAV orthomosaic generation methods in terms of online capability, geographic consistency, and multi-UAV scalability. Compared with traditional SfM/MVS-based pipelines, which typically require offline batch processing and substantial computational resources, our method enables incremental orthomosaic generation by continuously optimizing a unified Gaussian representation. Compared with existing Gaussian-based SLAM approaches that mainly focus on single-agent reconstruction, the proposed framework further considers multi-UAV collaboration by integrating GNSS-prior alignment and inter-agent visual loop refinement to establish a geo-consistent global map. The framework uses consumer-grade GNSS for initial geo-referenced alignment, while inter-agent visual loop closures refine moderate positioning errors; however, excessive GNSS errors may degrade global alignment, and complete GNSS denial is beyond the scope of this work. Meanwhile, the plane-guided collaborative densification and visibility-aware pruning strategies improve reconstruction

completeness in weakly constrained regions while controlling model complexity during incremental mapping.

Despite the promising results, the proposed method still has several limitations. First, the current positioning accuracy of approximately 2–3 m is not survey-grade; the system is intended for rapid online orthomosaic generation rather than high-precision surveying. Second, the spatial resolution of the generated orthomosaic is lower than that of traditional image stitching or texture-based methods, due to the inherent representation and rendering characteristics of 3D Gaussian Splatting. Third, the current plane-guided collaborative densification strategy assumes a dominant ground plane. For mountainous regions or scenes with significant elevation variations, a single-plane representation may introduce distortions in Gaussian initialization and affect the subsequent mapping quality.

In future work, we plan to address these limitations from three aspects.To improve absolute positioning accuracy, we will investigate higher-precision geographic constraints, such as RTK-GNSS and GCP-assisted optimization, together with tighter visual–GNSS fusion. To enhance orthomosaic resolution, we will investigate hybrid representations and rendering strategies that combine Gaussian-based modeling with high-frequency texture refinement, aiming to achieve both online performance and high-quality detail reconstruction. Furthermore, to extend the applicability of the proposed framework to complex terrains, we will investigate terrain-adaptive geometric modeling strategies, such as multi-plane representations or elevation-aware projection models, to overcome the limitations of the current dominant ground-plane assumption.

6. Conclusions

In this paper, we present a online incremental orthomosaic generation framework based on 3D Gaussian Splatting SLAM for multi-UAV aerial mapping. By integrating GNSS-prior alignment with inter-agent visual loop refinement, the proposed method achieves geo-consistent multi-UAV fusion within a unified global representation. To further enhance reconstruction quality and scalability, we introduce a plane-guided collaborative densification strategy to improve performance in under-constrained regions, as well as a visibility-aware importance pruning mechanism to control model complexity and computational cost. Extensive experiments on multiple UAV datasets demonstrate that the proposed method achieves competitive visual fidelity and geometric consistency compared to state-of-the-art baselines, while maintaining efficient runtime and bounded GPU memory usage.

Author Contributions: Conceptualization, X.Z. (Xiao Zhang) and S.L.; methodology, X.Z. (Xiao Zhang); software, X.Z. (Xiao Zhang); validation, X.Z. (Xiaozhou Zhu) and H.D.; formal analysis, X.Z. (Xiaozhou Zhu); investigation, X.Z. (Xiao Zhang); resources, S.L. and B.D.; data curation, X.Z. (Xiao Zhang) and H.Z.; writing—original draft preparation, X.Z. (Xiao Zhang); writing—review and editing, S.L. and B.D.; visualization, X.Z. (Xiao Zhang); supervision, S.L. and B.D.; project administration, S.L.; funding acquisition, S.L. and B.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data supporting the findings of this study are partially publicly available. The NPU-DroneMap dataset can be accessed from its official repository. The self-collected UAV datasets used in this study are not publicly available due to project restrictions, but may be made available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
SLAM	Simultaneous Localization and Mapping
3DGS	3D Gaussian Splatting
GNSS	Global Navigation Satellite System
ENU	East-North-Up
SfM	Structure from Motion
MVS	Multi-View Stereo
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure

References

1. Messaoudi, K.; Oubbati, O.S.; Rachedi, A.; Lakas, A.; Bendouma, T.; Chaib, N. A survey of UAV-based data collection: Challenges, solutions and future perspectives. *J. Netw. Comput. Appl.* **2023**, *216*, 103670. [CrossRef]
2. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97. [CrossRef]
3. Remondino, F.; Barazzetti, L.; Nex, F.; Scaioni, M.; Sarazzi, D. UAV photogrammetry for mapping and 3d modeling–current status and future perspectives. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *38*, 25–31. [CrossRef]
4. Zhang, J.; Xu, S.; Zhao, Y.; Sun, J.; Xu, S.; Zhang, X. Aerial orthoimage generation for UAV remote sensing. *Inf. Fusion* **2023**, *89*, 91–120. [CrossRef]
5. Ren, X.; Wei, X.; Li, Z.; Fu, Y.; Zhang, Y.; Xue, X. Deepsfm: Robust deep iterative refinement for structure from motion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *46*, 4058–4074. [CrossRef] [PubMed]
6. Wang, F.; Zhu, Q.; Chang, D.; Gao, Q.; Han, J.; Zhang, T.; Hartley, R.; Pollefeys, M. Learning-based multi-view stereo: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2026**, *48*, 6241–6260 [CrossRef] [PubMed]
7. Agisoft LLC. *Agisoft Metashape Professional*, version 2.0; Agisoft: St. Petersburg, Russia, 2023. Available online: https://www.agisoft.com (accessed on 5 February 2026).
8. Bentley Systems. *ContextCapture*, version 23.00.00.1317; Bentley Systems, Incorporated: Exton, PA, USA, 2023.
9. Pix4D SA. *Pix4Dmapper Photogrammetric Software*, version 4.8; Pix4D SA: Lausanne, Switzerland, 2023. Available online: https://www.pix4d.com (accessed on 5 February 2026).
10. Bu, S.; Zhao, Y.; Wan, G.; Liu, Z. Map2DFusion: Real-time incremental UAV image mosaicing based on monocular SLAM. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 4564–4571. [CrossRef]
11. Li, R.; Gao, P.; Cai, X.; Chen, X.; Wei, J.; Cheng, Y.; Zhao, H. A Real-Time Incremental Video Mosaic Framework for UAV Remote Sensing. *Remote Sens.* **2023**, *15*, 2127. [CrossRef]
12. Kern, A.; Bobbe, M.; Khedar, Y.; Bestmann, U. OpenREALM: Real-time mapping for unmanned aerial vehicles. In *Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS)*; IEEE: Piscataway, NJ, USA, 2020; pp. 902–911.
13. Zhang, X.; Dong, H.; Zhang, H.; Zhu, X.; Li, S.; Deng, B. Efficient Large-Scale Orthomosaicking: Integrating Learning-Based Visual Odometry with GNSS for Real-Time Consistency. *IEEE Trans. Geosci. Remote Sens.* **2026**, *64*, 3000117. [CrossRef]
14. Kerbl, B.; Kopanas, G.; Leimkühler, T.; Drettakis, G. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **2023**, *42*, 139-1–139-14. [CrossRef]
15. Feng, D.; Chen, Z.; Yin, Y.; Zhong, S.; Qi, Y.; Chen, H. CaRtGS: Computational Alignment for Real-Time Gaussian Splatting SLAM. *IEEE Robot. Autom. Lett.* **2025**, *10*, 4340–4347. [CrossRef]
16. 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 21584–21593.
17. Yugay, V.; Li, Y.; Gevers, T.; Oswald, M.R. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv* **2023**, arXiv:2312.10070.
18. 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 21357–21366.
19. Wang, X.; Zhang, W.; Xie, H.; Ai, H.; Yuan, Q.; Zhan, Z. Tortho-gaussian: Splatting true digital orthophoto maps. *arXiv* **2024**, arXiv:2411.19594.

20. Wei, J.; Zhu, G.; Chen, X. NeRF-based large-scale urban true digital orthophoto map generation method. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *18*, 1070–1084. [CrossRef]
21. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
22. 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 Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
23. Wang, R.; Schworer, M.; Cremers, D. Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3903–3911.
24. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
25. Kazerouni, I.A.; Fitzgerald, L.; Dooly, G.; Toal, D. A survey of state-of-the-art on visual SLAM. *Expert Syst. Appl.* **2022**, *205*, 117734. [CrossRef]
26. Qin, T.; Pan, J.; Cao, S.; Shen, S. A general optimization-based framework for local odometry estimation with multiple sensors. *arXiv* **2019**, arXiv:1901.03638.
27. Cao, S.; Lu, X.; Shen, S. GVINS: Tightly coupled GNSS–visual–inertial fusion for smooth and consistent state estimation. *IEEE Trans. Robot.* **2022**, *38*, 2004–2021. [CrossRef]
28. Wang, X.; Li, X.; Yu, H.; Chang, H.; Zhou, Y.; Li, S. GIVL-SLAM: A robust and high-precision SLAM system by tightly coupled GNSS RTK, inertial, vision, and LiDAR. *IEEE/ASME Trans. Mechatron.* **2024**, *30*, 1212–1223. [CrossRef]
29. Schmuck, P.; Chli, M. CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. *J. Field Robot.* **2019**, *36*, 763–781. [CrossRef]
30. Schmuck, P.; Ziegler, T.; Karrer, M.; Perraudin, J.; Chli, M. Covins: Visual-inertial slam for centralized collaboration. In *Proceedings of the 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*; IEEE: Piscataway, NJ, USA, 2021; pp. 171–176.
31. Tian, Y.; Chang, Y.; Arias, F.H.; Nieto-Granda, C.; How, J.P.; Carlone, L. Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems. *IEEE Trans. Robot.* **2022**, *38*, 2022–2038. [CrossRef]
32. Chen, Q.; Li, G.; Xue, X.; Pu, J. Multi-lio: A lightweight multiple lidar-inertial odometry system. In *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2024; pp. 13748–13754.
33. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **2021**, *65*, 99–106.
34. Hanson, A.; Tu, A.; Singla, V.; Jayawardhana, M.; Zwicker, M.; Goldstein, T. PUP 3D-GS: Principled Uncertainty Pruning for 3D Gaussian Splatting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 10–17 June 2025; pp. 5949–5958.
35. Hanson, A.; Tu, A.; Lin, G.; Singla, V.; Zwicker, M.; Goldstein, T. Speedy-splat: Fast 3d gaussian splatting with sparse pixels and sparse primitives. In Proceedings of the Computer Vision and Pattern Recognition Conference, Nashville, TN, USA, 10–17 June 2025; pp. 21537–21546.
36. Liu, W.; Guan, T.; Zhu, B.; Xu, L.; Song, Z.; Li, D.; Wang, Y.; Yang, W. Efficientgs: Streamlining gaussian splatting for large-scale high-resolution scene representation. *IEEE MultiMed.* **2025**, *32*, 61–71. [CrossRef]
37. Chen, Y.; Lee, G.H. Dogs: Distributed-oriented gaussian splatting for large-scale 3d reconstruction via gaussian consensus. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 34487–34512. [CrossRef]
38. Liu, Y.; Luo, C.; Fan, L.; Wang, N.; Peng, J.; Zhang, Z. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2024; pp. 265–282.
39. Liu, Y.; Luo, C.; Mao, Z.; Peng, J.; Zhang, Z. Citygaussianv2: Efficient and geometrically accurate reconstruction for large-scale scenes. *arXiv* **2024**, arXiv:2411.00771.
40. Yan, C.; Qu, D.; Xu, D.; Zhao, B.; Wang, Z.; Wang, D.; Li, X. Gs-slam: Dense visual slam with 3d gaussian splatting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 19595–19604.
41. Deng, T.; Chen, Y.; Zhang, L.; Yang, J.; Yuan, S.; Liu, J.; Wang, D.; Wang, H.; Chen, W. Compact 3d gaussian splatting for dense visual slam. *arXiv* **2024**, arXiv:2403.11247.
42. Ha, S.; Yeon, J.; Yu, H. Rgbd gs-icp slam. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2024; pp. 180–197.
43. 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. In *Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2024; pp. 10159–10166.
44. Zhu, S.; Qin, R.; Wang, G.; Liu, J.; Wang, H. Semgauss-slam: Dense semantic gaussian splatting slam. In *Proceedings of the 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2025; pp. 21174–21181.

45. Ji, Y.; Liu, Y.; Xie, G.; Ma, B.; Xie, Z.; Liu, H. Neds-slam: A neural explicit dense semantic slam framework using 3d gaussian splatting. *IEEE Robot. Autom. Lett.* **2024**, *9*, 8778–8785. [CrossRef]
46. Hong, S.; He, J.; Zheng, X.; Zheng, C. LIV-GaussMap: LiDAR-inertial-visual fusion for real-time 3D radiance field map rendering. *IEEE Robot. Autom. Lett.* **2024**, *9*, 9765–9772. [CrossRef]
47. Yang, J.; Cai, Z.; Wang, T.; Ye, T.; Gao, H.; Huang, H. Ortho-3DGS: True Digital Orthophoto Generation From Unmanned Aerial Vehicle Imagery Using the Depth-Regulated 3D Gaussian Splatting. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 10972–10994. [CrossRef]
48. Gao, G.; Yuan, M.; Ma, Z.; Gu, J.; Meng, W.; Xu, S.; Zhang, X. GeoROS: Georeferenced Real-time Orthophoto Stitching with Unmanned Aerial Vehicle. In *Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2022; pp. 2250–2256.
49. Jiang, C.; Zhou, X.; Chen, H.; Liu, T. UAV Positioning Using GNSS: A Review of the Current Status. *Drones* **2026**, *10*, 91. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.