The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-1-2024
ISPRS TC I Mid-term Symposium “Intelligent Sensing and Remote Sensing Application”, 13–17 May 2024, Changsha, China

Dense collaborative mapping with deep visual SLAM method

Peiwei Pan, Wei Zhang, Norbert Haala

Institute for Photogrammetry, University of Stuttgart, Germany
st175937@stud.uni-stuttgart.de, wei.zhang@ifp.uni-stuttgart.de, norbert.haala@ifp.uni-stuttgart.de

Keywords: Collaborative Mapping, Visual SLAM, Bundle Adjustment, Deep Learning, Optical Flow

Abstract

The creation of highly accurate and collaborative mapping algorithms is crucial for the progress of SLAM technology, as it greatly improve the efficiency of building detailed maps. In the area of mapping based on single moving trajectory, DROID-SLAM (Differentiable Recurrent Optimization-Inspired Design) by (Teed and Deng, 2021) stands out as an innovative method based on deep learning, providing a visual-only solution that works with various types of camera, such as monocular, stereo, and RGB-D. Its ability to create maps with excellent accuracy makes it superior to well-known methods like ORB-SLAM3 by (Campos et al., 2021). Despite its impressive individual mapping performance, DROID-SLAM does not account for scenarios involving multi-session data or the collaborative map creation by multiple agents. To address this problem, we propose two collaborative map construction algorithms built upon DROID-SLAM. Compared to prior methods that compute explicit relative transformations for loop closures, our algorithm leverages the power of deep learning-based bundle adjustment, using dense per-pixel correspondence, to merge into a globally consistent state. These algorithms have been thoroughly tested with stereo and RGB-D models. we validated the effectiveness of our proposed algorithms on both public and self-collected datasets, showing higher accuracy than prior methods. By leveraging the strengths of DROID-SLAM while addressing its limitations with our novel algorithms, we extend the application scenarios of this method and provide a new way of thinking about collaborative mapping.

1. Introduction

Simultaneous Localization and Mapping (SLAM) is crucial for robotics and augmented reality, enabling devices to map their environment and locate themselves within it. Among the various approaches in SLAM, Visual SLAM and LiDAR SLAM stand out as prominent options. Visual SLAM is more cost-effective and has become widely adopted. However, it presents challenges, as depth is inferred rather than directly measured, unlike in LiDAR SLAM, where depth measurement is direct and precise. Despite this, Visual SLAM captures rich visual information, offering valuable data that LiDAR SLAM does not. In our research, we focus on employing dense Visual SLAM as our foundational technique. This choice allows us to leverage the comprehensive visual information it provides. Recent advancements focus on expanding SLAM's capabilities for larger, more complex areas through multi-session and collaborative mapping. Multi-session SLAM merges maps from separate sessions into one, ideal for large areas or when single-session mapping is limited by factors like time or battery life. Collaborative SLAM, involving multiple agents mapping together, accelerates the process and improves map quality, requiring advanced coordination and data sharing.

Collaborative mapping relies on the foundation of single trajectory mapping, making the choice of a precise algorithm essential. ORB-SLAM3, by extracting and tracking ORB features from image sequences, estimates camera motion and optimizes mapping, yet struggles in environments lacking distinct textures. SVO by (Forster et al., 2017), employing direct pixel intensity for tracking and optimizing camera motion, faces accuracy issues in feature-sparse scenes. LSD-SLAM by (Engel et al., 2014) adopts a direct approach, using pixel intensities for mapping and motion estimation without feature point extraction, but as a monocular system, it struggles with scale consistency over time due to its inability to gauge absolute scale from a single image.

DROID-SLAM enhances visual SLAM with deep learning, enabling robust camera pose tracking through advanced feature extraction from entire images, which proves effective even in environments lacking texture, can well address these shortcomings. The system supports a variety of camera types for versatile application. Deep neural networks are integrated to predict dense optical flow correspondences. Representative keyframes with sufficient baseline margins are selected for algorithm efficiency based on average flow distance. Additionally, the network predicts per-pixel uncertainty metrics to mitigate the impact of outlier correspondences. These metrics are further utilized by our collaborative mapping algorithm to identify confident overlapped areas between submaps.

Despite its advantages, the original DROID-SLAM lacked collaborative mapping capabilities. Building on its foundation and leveraging its strengths, we introduce two collaborative mapping methods integrated into our novel ”Micro DROID Loop” concept. Each method, known for its efficiency and robustness, can be chosen based on the extent of the common viewing areas. We've expanded the original algorithm, previously limited to single trajectory mapping, to support collaborative mapping. This enhancement allows our collaborative mapping to surpass even more robust and precise multi-sensor fusion collaborative mapping algorithms in performance and also offers a new approach for high-precision collaborative mapping in the vision domain. Specifically, there are the following:

- The Micro DROID Loop concept is introduced as a fundamental unit in which two segments of keyframes within a common visibility area are interconnected. First these two sequences are connected in an optimal way. Then the poses of the reference keyframes are fixed in their own local coordinate system, while the poses of the keyframes

to be aligned are set as unknown. Subsequently, by executing the DROID-SLAM algorithm within this connected sequence of keyframes, the new poses of the keyframes to be aligned are determined. This process results in the keyframes to be aligned having two sets of poses: the poses in their own local coordinate system and new poses that are aligned with the reference keyframes within the same coordinate system.

- When there is a substantial overlap between maps, we employ the more time-efficient Method 1 within the Micro DROID Loop unit. This simply requires calculating the transformation matrix between the own local coordinate system and new poses of the keyframes to be aligned in the overlapping area, and then applying this transformation matrix to all keyframes to be aligned throughout the entire area. Due to the use of a deep learning-based camera relocation coordinate calculation method, the accuracy of the relocation coordinates for the new poses is improved, so the obtained transformation results are more stable.

- When the overlapping area between maps is small, focusing solely on high time-efficiency local pose transformations can lead to local optimizations that fail globally, resulting in alignment failure between maps. Therefore, we propose the more robust but less time-efficient Method 2. In the Micro DROID Loop, we innovatively use the deep learning-based optical flow from the DROID-SLAM frontend to introduce a method for calculating new poses by directly stitching two segments of keyframes together, entirely eliminating the need for calculating transformation matrices. Finally, we perform a backend optimization of DROID-SLAM on the stitched sequence of keyframes to be aligned.

To delve deeper into the two new collaborative mapping methods proposed for DROID-SLAM in this paper, we will first briefly introduce related works. This includes the process of single trajectory mapping with DROID-SLAM, as understanding this process is crucial since collaborative mapping builds upon it. To efficiently connect all sub-maps two by two, we will introduce Prim’s algorithm by by (Prim, 1990). We also cover image matching algorithms used in the collaborative mapping process, as quickly and efficiently identifying shared visibility areas is essential. Additionally, we discuss multi-sensor fusion collaborative mapping algorithms to compare their results with our visual collaborative mapping outcomes, demonstrating the superiority of our approach. Next, we will meticulously outline the entire collaborative algorithm process, with a focus on maximizing the advantages of the original DROID-SLAM algorithm. Finally, we will conduct two experiments to validate our proposed methods: one based on a public dataset assessing the overall pose sequence of the camera, and another using a self-collected dataset to evaluate the precision of the point clouds generated by collaborative mapping.

2. Related Works

Using DROID-SLAM, high-precision construction of sub-maps for single trajectories can be achieved. During the process of submap construction, video streams are utilized as input for real-time reconstruction and localization. The frontend handles frames, extracts features based on deep learning with optical flow, selects keyframes, and performs local bundle adjustment, while the backend conducts global bundle adjustment on the history of keyframes. Within the frontend process, upon initialization completion, a frame graph with edges between initialized keyframes within 3 adjacent frames is established. The frontend directly operates on the video stream, maintaining keyframes and a frame graph with visually connected keyframe edges. The poses and depths of keyframes are actively optimized. The backend utilizes the distance matrix between keyframes formed by optical flow based on deep learning to conduct global bundle adjustment and indirectly realizing the functionality of loop detection. So we have obtained the coordinates of each sub-map within its own local coordinate system before doing the collaborative map building.

The order of connecting sub-maps pairwise is crucial, as a greater overlap area implies higher precision achievable in collaborative mapping. To address this sequencing challenge, we employ Prim’s algorithm, a method for finding the minimum spanning tree in a weighted undirected graph. A minimum spanning tree is a tree structure that includes all the vertices of the graph, with the sum of the weights of its edges being the smallest possible. The core idea behind Prim’s algorithm is to start from a random vertex and progressively add edges and vertices to the tree until it encompasses all vertices in the graph. We can abstract the initial reference map as a starting random point, the other sub-maps to be aligned as additional vertices, and the degree of overlap between maps as the weight. The higher the overlap, the lower the weight.

To expedite the identification of approximate common viewing areas between keyframe sequences of different sub-maps, we prioritize time efficiency over high precision in matching. This approach allows for the initial detection of common viewing areas, which, upon confirmation, are meticulously connected through our proposed collaborative mapping algorithm. We utilize the FBOW (Fast Bag of Words) by part of (Muñoz-Salinas and Medina-Carnicer, 2020) algorithm, an advanced iteration of the conventional Bag of Words model by (Zhang et al., 2010), designed for faster processing and reduced memory usage while maintaining effective matching capabilities. In contrast, while high-precision image matching algorithms like FV(Fisher Vectors) by (Klein et al., 2015) offer greater computational complexity and accuracy, their extensive processing time makes them less suitable for the preliminary phase of common viewing area detection.

By comparing the accuracy of our visual collaborative mapping approach with those based on multi-sensor fusion, we demonstrate its superiority even with fewer sensors. We examine three representative multi-sensor fusion collaborative mapping algorithms: ORB-SLAM3 integrates IMU and traditional camera setups for robust pose estimation and higher map precision in collaborative settings; COVINS by (Schmuck et al., 2021) focuses on visual-inertial collaboration, excelling in processing data from cameras and IMU across networked scenes with efficient data sharing and optimization for scalable, accurate mapping; and MAPLAB 2.0 by (Cramariuc et al., 2022), which, besides cameras and IMU, incorporates LiDAR data for enhanced environmental modeling and navigation solutions crucial in complex mapping tasks. Despite their integration of multiple sensors for high mapping accuracy, our experiments show that our camera-only collaborative mapping surpasses them.

3. Methodology

In this section, we will first provide an overview to our collaborative mapping process in Section 3.1, followed by a detailed ex-

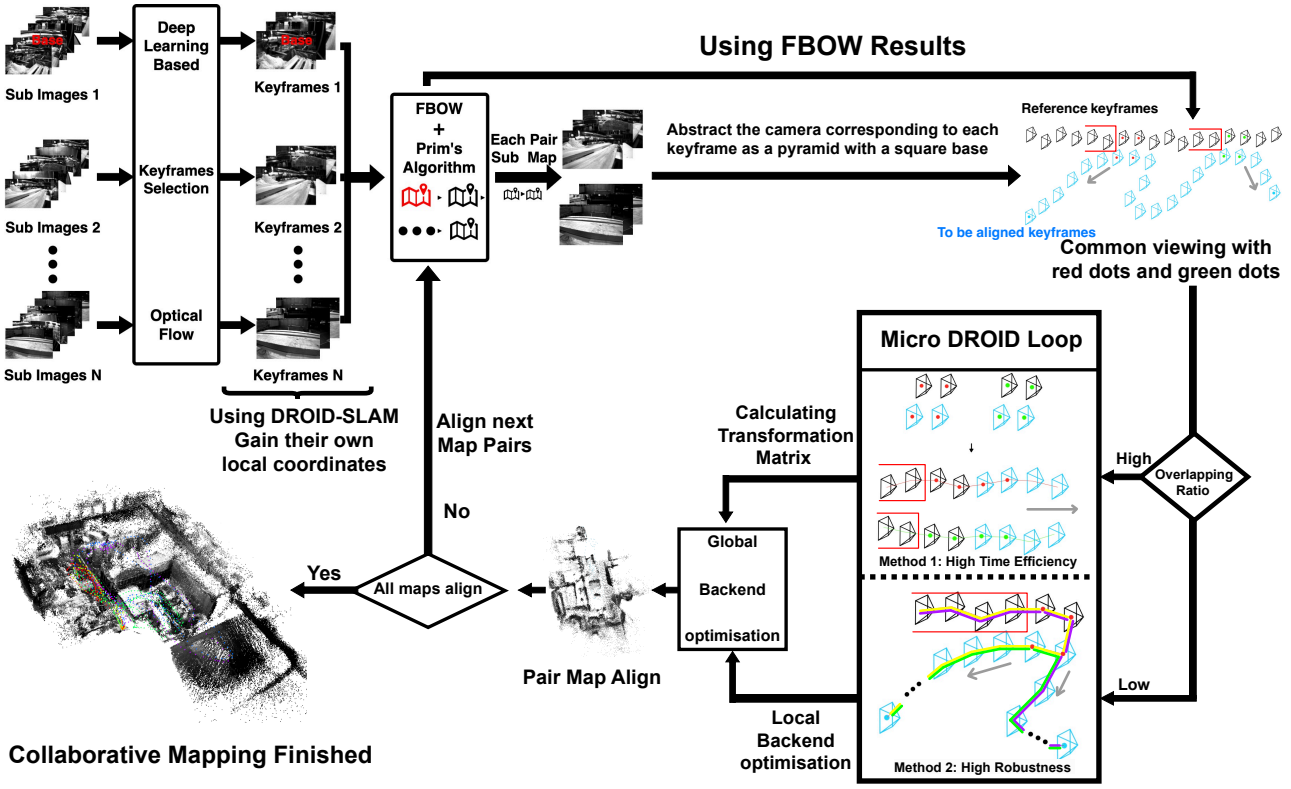

Figure 1. Flowchart of the collaborative mapping process.The entire flowchart begins with keyframe extraction in the top left corner. The red box represents the area extended in the reverse direction and the gray arrow represents the possible direction of the forward extension. In method 2, the yellow and purple line segments represent the two keyframe sequences that were recombined, and the green represents the part that needs local backend optimization later.

planation of the submap linking algorithm in Section 3.2, which identifies overlap keyframes for linking submaps. Next, in Section 3.3 and Section 3.4, we validate linked candidates before passing them to the core of our algorithms for submap merging, as outlined in Section 3.5. Finally, in Section 3.6, we utilize global backend optimization to refine overall consistency.

3.1 Overview

In Figure 1, the stages of the collaborative map building process are depicted, starting with the extraction of keyframes from image sequences.This step is crucial as it significantly reduces the data volume for collaborative mapping while ensuring its quality, by focusing solely on keyframes. Leveraging the considerable advantages of DROID-SLAM’s use of deep learning-based optical flow for keyframe selection, we can minimize the keyframes that require processing to the greatest extent.

After extracting keyframes, using Prim’s algorithm and FBOW algorithm to connect the sub-maps sequentially based on the degree of similarity between them. For each two submaps to be aligned together, extracting common viewing within them. Then the merging strategy is determined based on the overlapping ratio between submaps. Specifically, if the overlap exceeds half of their area, the process employs method 1, as shown in the upper half of the Micro DROID Loop unit in figure 1. This method calculates the pose transformation matrix, which is then applied to all keyframes of the submap to be aligned, aligning it with the other reference submap. If the overlap is less than half, method 2 is applied instead. Illustrated in the lower half of figure 1, this method directly calculates the new poses for the submap to be aligned in the reference coordinate system of reference submap, bypassing the need for a transformation matrix. But this method needs to do the local backend optimisation later for the whole submap to be aligned with new poses. This approach is generally used when the spatial relationship between the submaps is less direct.

After choosing the appropriate method, global backend optimization which is the same as the backend process using DROID-SLAM method in two submaps construction plays a crucial role. This optimization process fine-tunes the merged map, ensuring that it is both coherent and accurate. It adjusts the poses of two submaps based on the overall structure, reducing errors and discrepancies. The iterative nature of this process means that after each pair of submaps is merged and optimized, the system checks if there are still unmerged submaps remaining. The procedure repeats, merging submaps and optimizing the global structure, until all have been integrated into a unified map.

The above describes the whole process of collaborative mapping, the following will delve deeper into the details of collaborative mapping.

3.2 Submap Links Sequences

To efficiently link submaps in a mapping system, we employ Prim’s algorithm with a unique approach, where each submap, represented by its keyframes, is considered a node. The process begins by selecting a base submap as the initial node, setting the stage for the linkage of subsequent submaps. Central to

this approach is the use of FBOW algorithm to detect overlaps between pairs of submaps. By calculating the overlap ratio, we assess the degree of visual similarity between submaps. This ratio is instrumental in identifying which submaps share significant portions of their views. We then use the reciprocal of this overlap ratio as the weight for the links between nodes. This method ensures that connections between submaps with higher visual similarities are prioritized, leading to a more intuitive linking strategy. Through Prim’s algorithm, we construct a minimum spanning tree from the base submap, linking all submaps in a way that minimizes the overall connection weight. This strategy achieves an effective and efficient linkage of submaps, ensuring that those with greater visual overlaps are closely connected. By implementing this method, we ensure a streamlined process for integrating submaps, resulting in a detailed and accurate map. At the same time, we can re-use the results of the FBOW algorithm in the later overlapping views matching.

3.3 Micro DROID Loop

The Micro DROID Loop refers to a sequence of reference keyframes and a sequence of to-be-aligned keyframes within the overlapping views, where the poses of the reference keyframes are known and initialized, and the poses of the to-be-aligned keyframe sequence are unknown. Since the algorithm starts from the reference keyframe sequence, in order to ensure that the poses of the to-be-aligned keyframe sequence is computed stably, we need to increase the number of reference keyframes inversely, in addition to the keyframes of the overlapping view, until the total number of reference keyframes in the Micro DROID Loop reaches a threshold of 20 (shown with reverse extension area in figure 1). If there are not enough reference keyframes then this overlapping view is discarded.

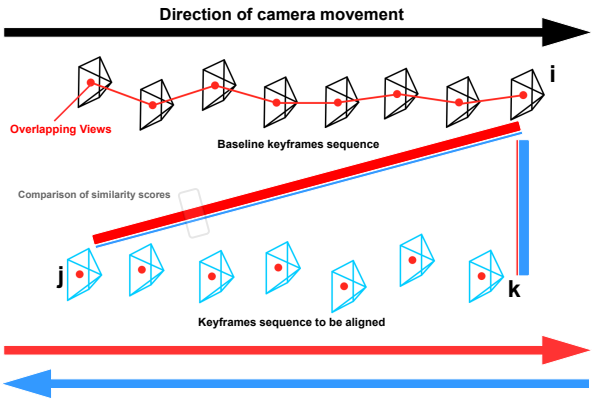

Figure 2. Principle of linking two sub-sequence keyframes

Because we need to combine two keyframes sequences, the way the two keyframe sequences are linked is very important, as shown in Figure 2. In any overlapping view of the two keyframe sequences, the last keyframe i of the reference keyframe sequence is picked, and also the first keyframe j and the last keyframe k of the to-be-aligned keyframe sequence are picked. the group with a higher score is selected by calculating the similarity scores of i-j, i-k using FBOW, and then the direction of advancement is decided by the picked j or k. The higher the score, the thicker the corresponding connecting line segment shown in Figure 2. Along this direction, we can somewhat increase the number of keyframes to be aligned without overlapping views (shown with Forward extension arrow direction in figure 1) to ensure that the backend process of this mini-loop is more stable.

3.4 Connection validity determination

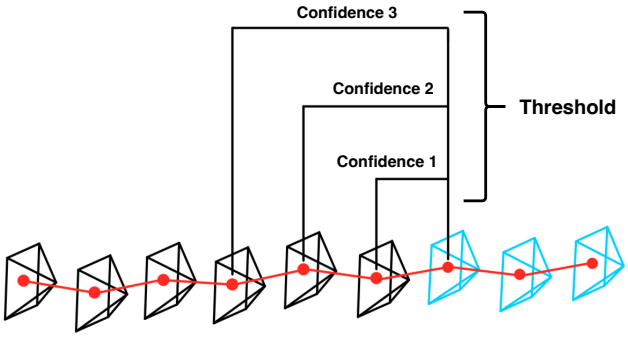

Figure 3. Compare the mean of three confidences with the threshold

Given that we determine camera poses using an optical flow method based on deep learning, the success certainty of the optical flow method is critical. After completing the link in each region with overlapping views, we need to judge whether the optical flow method at the link of two keyframe sequences is successful. The error function of DROID-SLAM is defined by minimizing the reprojection error among multiple pairs of keyframes within a frame window and assigns a suitable confidence to each pair of keyframes through deep learning. We use this confidence to determine the success of the optical flow method at the link part. As shown in Figure 3, if the average confidence of the last three frames of the reference keyframe sequence and the first frame of the to-be-aligned keyframe sequence exceeds a certain threshold, then the optical flow method is deemed to be successful, meaning the link is successful. Such a success signifies not only the effectiveness of the optical flow in this context but also ensures the integrity and reliability of the connection established between these overlapping views.

3.5 Two Collaborative mapping methods

In order to clearly state our algorithm, we list the pseudo-code of the algorithm 1. We then describe the algorithmic process and explain the superiority of our algorithm in a textual manner with figure 1.

The upper half of Micro DROID Loop unit in Figure 1 describes the principle of the first method of collaborative mapping, which is particularly effective for submaps with a high number of overlapping view regions. For two submaps composed of sequences of keyframes that need to be aligned, the results of FBOW algorithm is utilized to identify all overlapping view regions. For each of these regions, the connection and extension methods described in the previous section are applied, followed by a determination of the connection’s success. If the connection proves successful, which means the optical flow method is effective, the poses of the keyframes to be aligned are treated as unknowns, while the poses of the reference keyframes are considered known quantities. These known poses are derived from the original locations obtained after mapping the submaps independently. However, if the reference map has previously been transformed into another coordinate system to ensure consistency, the coordinates of the transformed coordinate system are used instead. Subsequently, a localized reconstruction of each successfully connected overlapping view region is performed. This process enables the sequence of keyframes within the overlapping region that awaits alignment to

Algorithm 1 Collaborative mapping medthods

S_1: submap with reference keyframes
S_2: submap with to-be-aligned keyframes
MDL: Micro DROID Loop
TRS: Transformation matrix calculation process
Ω, r = FINDOVERLAPSUSEFBOW(S_1, S_2)
if $r > overlapThres$ **then**
    Method 1:
    $T_{overlaps} = \{\}$
    for each $o_i \in \Omega$ **do**
        o_i^{ex} = EXTEND(o_i)
        c = COMPUTECONFIDENCE(o_i^{ex})
        if $c > confidenceThres$ **then**
            $newS_2^{ex}$= MDL(o_i^{ex})
            T_i = TRS($newS_2^{ex}, S_2^{ex}$)
            $T_{overlaps} = T_{overlaps} \cup T_i$
        end if
    end for
    T_{best} = QUARTILECALCULATION($T_{overlaps}$)
    $S_2^{aligned}$ = TRANSFORMATION(T_{best}, S_2)
else
    Method 2:
    o_{best} = CONFIDENCESELECT(Ω)
    o_{best}^{ex} = EXTEND(o_{best})
    $S_2^{part1} \cup S_2^{part2} \leftarrow S_2$
    $newSeq_1 \leftarrow o_{best}^{ex}[S_1] \cup S_2^{part1}$
    $newSeq_2 \leftarrow o_{best}^{ex}[S_1] \cup S_2^{part2}$
    $newS_2^{part1}$ = MDL($newSeq_1$)
    $newS_2^{part2}$ = MDL($newSeq_2$)
    $newS_2 \leftarrow newS_2^{part1} \cup newS_2^{part2}$
    $S_2^{aligned}$ = LOCALBACKEND($newS_2$)
end if
GLOBALBACKEND($S_1, S_2^{aligned}$)

acquire coordinates under the new coordinate system. These coordinates, along with the original coordinates obtained from independent mapping, are then used to calculate the transformation matrix. Finally, applying the derived transformation matrix to all the keyframe sequences that need to be aligned accomplishes the preliminary alignment. This method only requires consideration of coordinate transformations within overlapping view regions, making it highly time efficient. However, if there are only a few overlapping view regions between two submaps, this approach may not yield good alignment results. This is because it can only ensure that the local map alignment is optimized. Even minor changes to the transformation matrix can lead to significant errors in the camera poses far from the connection points. These substantial errors cannot be resolved through simple backend optimization, rendering this method less robust. However, between submaps with high overlapping areas, this method is very time efficient, which is the biggest advantage of this method.

With fewer overlapping ratio, we have proposed a second algorithm that is significantly more robust, as illustrated in the lower half of Micro DROID Loop unit in Figure 1. However, compared to the first method, this approach is not as time efficient because it involves considering all keyframes of the map to be aligned and it needs backend optimisation one more time. Initially, we select the segment with the best optical flow connection effect among all overlapping view regions, specifically the overlapping region in the reference keyframe sequence with the highest average confidence for the last three keyframes. Next, after determining the direction of the connection as described in sec 3.3, we create two new sequences as shown in the lower half of the Micro DROID Loop unit in Figure 1 with yellow and purple lines. One new sequence includes the reference keyframe containing the common-view region extended in the reverse direction with the keyframes to be aligned along the connection direction up to the first or last sheet of the entire segment of keyframes to be aligned; the other new sequence likewise includes the reference keyframe containing the common-view region extended in the reverse direction with the remaining keyframes to be aligned up to the last or first sheet of the keyframes to be aligned. The poses of the reference keyframes are given initial values, while the poses of the keyframes to be aligned are set as unknowns. Through the reconstruction of these two keyframe sequences into submaps, we can obtain the coordinates of all keyframes to be aligned in the new coordinate system without the need for coordinate transformation. Finally, local backend optimisation is performed on all keyframes where the new pose is obtained (shown in Figure 1 with the green line). This method significantly enhances the robustness of the alignment process by leveraging the optical flow connections with the highest effectiveness and extending the keyframe sequences in a manner that ensures a comprehensive and accurate alignment.

3.6 Global backend optimisation

Global backend optimisation: We integrate the two keyframe sequences for a comprehensive backend optimization. Through the previous steps, the initial poses of these two keyframe sequences are unified within the same coordinate system. The process of whole optimization directly employs the backend component utilized during the independent mapping phase. By leveraging the backend optimization processes established in the initial mapping, we are able to refine the alignment and poses of the keyframes, addressing potential inconsistencies and improving the precision of the reconstructed environment. This methodological continuity reinforces the integrity of the mapping process, enabling a more seamless and accurate integration of keyframe sequences into the comprehensive environmental model.

4. Experiment

We begin by validating the complementary nature of our two proposed collaborative mapping methods (Sec 4.2), followed by quantitative evaluations of camera pose accuracy using the public EuRoC dataset (Sec 4.3). We then conclude with the evaluation of the merged dense point cloud map using our self-collected dataset (Sec 4.4).

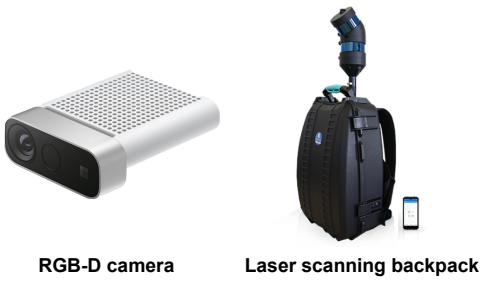

Figure 4. Personal dataset collection devices

4.1 Datasets preparation

To assess two collaborative mapping algorithms, we tested them with the EuRoC MAV Dataset ((Burri et al., 2016)) for public

data, focusing on its Stereo mode, and a self-collected dataset using Azure Kinect DK and Heron’s LiDAR for personal data. The public dataset’s diverse conditions and provided ground truths allowed for precise algorithm evaluation. For the personal dataset, we compared LiDAR’s detailed point clouds against the camera-generated RGB point clouds, overcoming the challenge of obtaining accurate camera poses. This dual-dataset approach effectively demonstrates the algorithms’ robustness and accuracy. The deviced used for our dataset is shown in the figure 4.

4.2 Complementary nature of two methods

In the first experiment, designed to validate the necessity of two distinct collaborative mapping methods, we exclusively employed the Machine Hall sequences from the EuRoC dataset for our evaluations. The Machine Hall dataset consists of five sequential image sets, labeled MH01 through MH05. Utilizing Fast Bag of Words (FBOW) for keyframe matching, we identified considerable overlapping areas between sequences MH01-MH03 and between MH04-MH05, while the overlap between MH03 and MH04 was notably very minimal, falling below our overlap threshold.

By comparing these calculated poses with the ground-truth data provided within the public dataset, we were able to compute the Root Mean Square Absolute Trajectory Error (RMS ATE) for both segments of the trajectories, serving as a measure of alignment accuracy.

To underscore the necessity and effectiveness of the two collaborative mapping methodologies, we executed collaborative mapping on various subsequence combinations, the results of which, including time efficiency and precision metrics, are detailed in Table 1.

Room	Machine Hall			
Sequences	MH01-MH03		MH01-MH05	
Evaluation Indicators	RMS ATE (cm)	Time Consuming (s)	RMS ATE (cm)	Time Consuming (s)
Method 1	**1.7**	**62**	55	**131**
Method 2	1.8	161	**3.6**	340

Table 1. Comparison of the performance of the two collaborative mapping methods on EuRoC dataset

From the data presented in Table 1, it’s evident that the localization accuracies of the collaborative mapping methods applied to MH01-MH03, areas with more significant overlap, are comparable. Nevertheless, Method 1 distinguishes itself with markedly higher time efficiency. This observation leads us to conclude that in situations where there is a high degree of map overlap, concentrating exclusively on the coordinate system transformations of these overlapping areas is not only logical but also markedly efficient.

However, when our analysis encompasses the entire sequence range of MH01-MH05, the accuracy dynamics undergo a significant shift. The sequences MH04 and MH05 demonstrate a substantially reduced overlap with the initial three sequences, leading to notable inaccuracies when solely relying on the coordinate system transformations of the overlapping areas. To counteract this, we employed the second approach to collaborative mapping, which involves integrating calculations for both overlapping and non-overlapping areas. This methodology not only achieved accuracy comparable to that of the MH01-MH03 sequences using either method but also proved to be more robust.

Yet, this robustness comes at a cost: the approach necessitates a near threefold increase in the time required for processing. This substantial increase in time expenditure is attributed to the need to incorporate a greater number of keyframes and to conduct an additional backend optimization process for each map

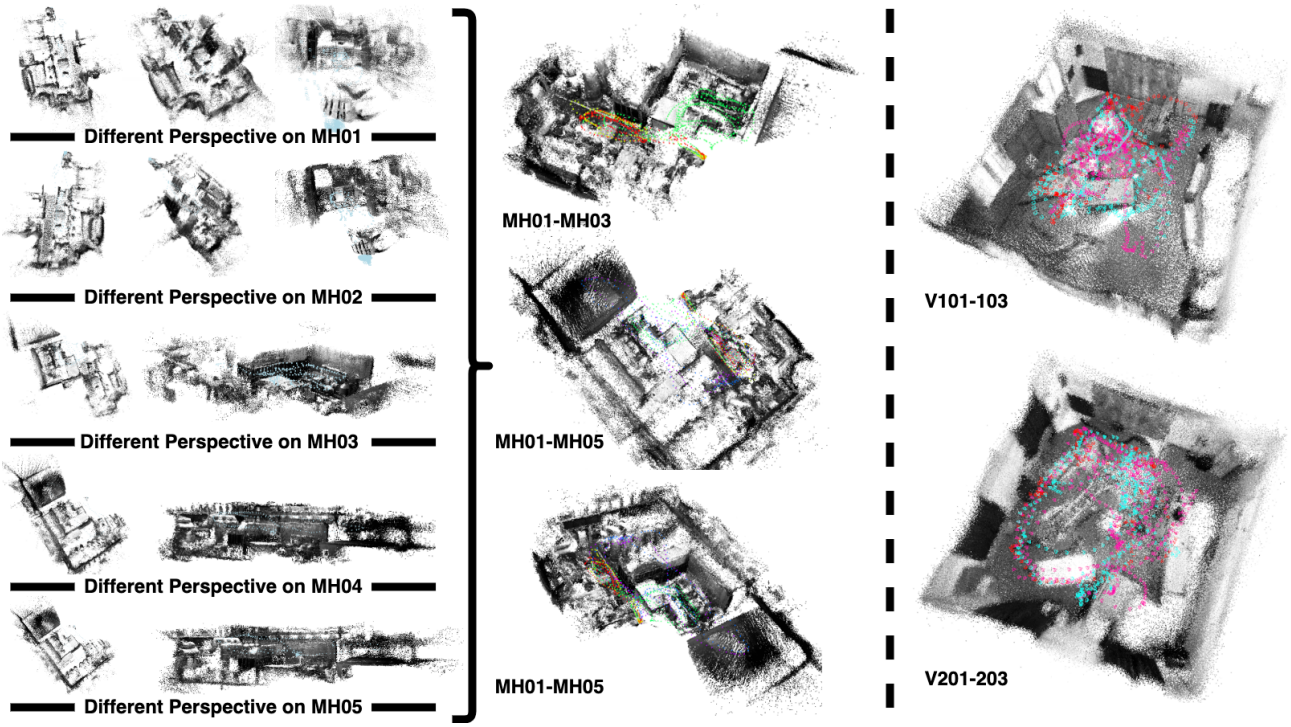

Figure 5. EuRoC dataset sub-maps collaborative mapping results. The different sequences of camera trajectories in the aligned map are represented by different coloured breakpoints

pairing. This finding indicates that while the inclusion of non-overlapping areas into the mapping process improves accuracy, it does so at the expense of significantly higher computational time and resources, highlighting a critical trade-off between accuracy and efficiency in collaborative mapping strategies.

4.3 Camera poses accuracy evaluation

In the second experiment, we evaluated the algorithmic accuracy of two methods by judging the poses accuracy of the camera during motion on the whole EuRoC dataset and compared it with the accuracy of other representative algorithms.

In assessing algorithmic accuracy, we first observe the results of map merging after applying appropriate methods, as shown in Figure 5. From left to right in Figure 5, the sub-map reconstruction results of the Machine Hall from different perspectives are initially presented, followed by the collaborative mapping reconstruction results for MH01-MH03 and the collaborative mapping reconstruction results from different perspectives for MH01-MH05. Finally, the collaborative mapping reconstruction results for two Vicon Rooms are shown. In all collaborative mapping reconstruction results, the continuous color breakpoints represent the camera trajectories of different sequences.

Subsequently, we quantitatively evaluated the accuracy of the algorithm. This was done by comparing the Root Mean Square Absolute Trajectory Error (RMS ATE) between the trajectories post-collaborative mapping and the ground-truth trajectories. We also compared these results with various outstanding algorithms, as shown in Table 2.

In our comparative analysis between our algorithm and other prominent algorithms such as COVINS, which integrates Inertial Measurement Units (IMUs) for enhanced tracking accuracy, the multi-sensor integrated platform Maplab 2.0 known for its versatility in handling various sensor inputs, and the well-regarded ORB-SLAM3 algorithm that is a benchmark in simultaneous localization and mapping (SLAM) technology, our findings indicate a significant improvement. Specifically, by relying exclusively on camera inputs, our algorithm demonstrates a universally higher precision in tracking and localization throughout the camera’s motion. This not only underlines the effectiveness of our approach but also highlights its superiority in scenarios where only camera data is available, proving its robustness and advanced capabilities in visual-based navigation and mapping tasks.

Room		Machine Hall		Vicon 1	Vicon 2
Sequences		MH01-03	MH01-05	V101-103	V201-203
ORB-SLAM3 Stereo	RMS ATE(cm)	2.8	4	**2.7**	16.3
COVINS	RMS ATE(cm)	2.4	**3.6**	4.2	-
Maplab 2.0	RMS ATE(cm)	4.3 (On average)			
Our Methods		Method 1	Method 2	Method 1	Method 1
Our Stereo	RMS ATE(cm)	**1.7**	**3.6**	3.1	**1.6**

Table 2. Comparison of the accuracy of various collaborative mapping algorithms on EuRoC dataset

4.4 Dense point cloud accuracy evaluation

The third experiment was designed to validate the accuracy of the dense point cloud after reconstruction. We utilized self-collected dataset, initially capturing images of a room divided into two areas with an RGB-D camera. As illustrated in Figures 6’s Area 1 and Area 2, these two areas are roughly separated by a bookshelf, with the shared view consisting of a portion of the wall’s clock as highlighted in Figure 6. It’s noticeable that these areas have limited shared views, allowing for the adoption of the second collaborative mapping approach for reconstruction. Subsequently, we employed a high-precision LiDAR integrated with a camera and IMU device to model the entire room, obtaining LiDAR point cloud data as shown in the lower

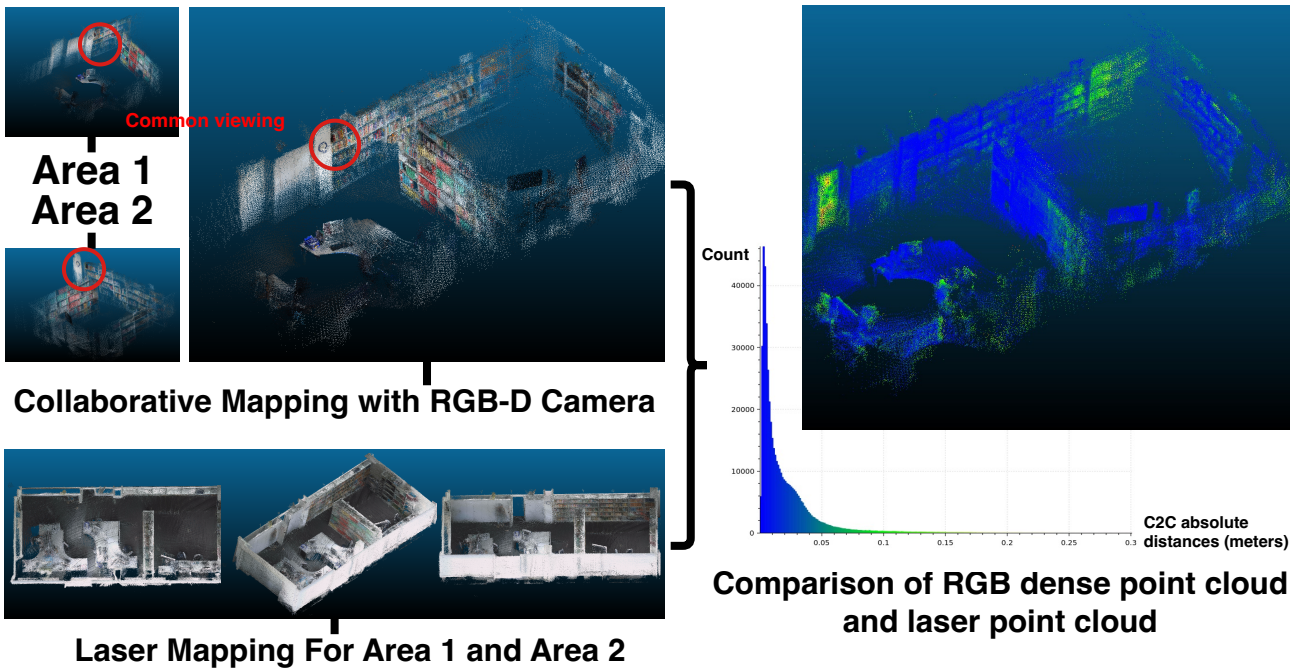

Figure 6. Comparison of the accuracy of collaborative mapping dense point cloud and laser point cloud with schematic and histogram. The schematic is in the top right corner and the histogram is in the bottom right corner, and the colours are shared between them.

part of Figure 6. Finally, we imported the point cloud data from the collaborative mapping and the LiDAR point cloud data into CloudCompare software. By manually selecting identical feature points for point cloud alignment and calculating the difference between the two sets of point cloud data, we obtained the results displayed in the right half of Figure 6.

From the analysis of the point cloud comparison depicted and histogram in the right half of Figure 6 , we observe a predominantly blue coloration in the point cloud generated through collaborative mapping. This blue hue signifies that the absolute distance discrepancy between the two compared point clouds is less than 5 centimeters, indicating a high degree of alignment between the point cloud produced by collaborative mapping and the one obtained from LiDAR scanning. Such a close match validates the efficiency of our reconstruction process, confirming that the collaborative mapping approach is not only viable but also effective in generating accurate spatial data.

Moreover, an interesting observation can be made regarding the point cloud representation of the door area, where a significant error is evident through the appearance of red points. This stark contrast arises due to the differing conditions under which the data was collected: the LiDAR scan was performed with the door open, whereas the RGB-D scan captured the door in a closed state. This discrepancy highlights the challenges faced in dynamic environments where changes in the scene between different scans can lead to substantial differences in the resulting data.

Further insights can be gleaned by examining the relationship between point cloud errors and their distance from the common viewing area. As the distance from this common viewpoint increases, the accuracy of the point cloud tends to decrease, leading to larger errors. This phenomenon is partly due to the inherent limitations in the precision of projecting 2D pixels onto a 3D space, which can introduce discrepancies of approximately 10 centimeters. Despite these challenges, the observed errors fall within a tolerable range, underscoring the robustness and practical applicability of our method.

5. Conclusion

In this paper, we introduce two collaborative mapping approaches tailored for visual SLAM, especially addressing scenarios with limited common viewing areas by proposing a viable solution. These methods have been rigorously validated through extensive experiments, proving to be both necessary and effective. Remarkably, even in scenarios reliant solely on camera inputs, our approaches achieve a higher collaborative mapping precision compared to other algorithms that integrate multiple sensors. This advancement underscores our methods' efficiency in leveraging visual data, setting a new benchmark for accuracy in the realm of collaborative visual SLAM.
In the future, we plan to incorporate the two proposed collaborative mapping algorithms into a deep learning framework, aiming for an end-to-end solution. This effort will potentially enhance mapping efficiency, paving the way for advanced autonomous systems and multi-robotics applications.

References

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W., Siegwart, R., 2016. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10), 1157–1163.

Campos, C., Elvira, R., Rodriguez, J. J. G., Montiel, J. M. M., Tardos, J. D., 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, 1–17.

Cramariuc, A., Bernreiter, L., Tschopp, F., Fehr, M., Reijgwart, V., Nieto, J., Siegwart, R., Cadena, C., 2022. maplab 2.0–a modular and multi-modal mapping framework. *IEEE Robotics and Automation Letters*, 8(2), 520–527.

Engel, J., Schöps, T., Cremers, D., 2014. Lsd-slam: Large-scale direct monocular slam. *European conference on computer vision*, Springer, 834–849.

Forster, C., Zhang, Z., Gassner, M., Werlberger, M., Scaramuzza, D., 2017. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Transactions on Robotics*, 33(2), 249-265.

Klein, B., Lev, G., Sadeh, G., Wolf, L., 2015. Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation.

Muñoz-Salinas, R., Medina-Carnicer, R., 2020. UcoSLAM: Simultaneous Localization and Mapping by Fusion of KeyPoints and Squared Planar Markers. *Pattern Recognition*, 107193.

Prim, R., 1990. Shortest connection networks and some generation. *Bell Systems Tech. J.*, 36, 368–379.

Schmuck, P., Ziegler, T., Karrer, M., Perraudin, J., Chli, M., 2021. Covins: Visual-inertial slam for centralized collaboration. *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, IEEE, 171–176.

Teed, Z., Deng, J., 2021. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*.

Zhang, Y., Jin, R., Zhou, Z.-H., 2010. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 1, 43–52.