2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

BAMF-SLAM: Bundle Adjusted Multi-Fisheye Visual-Inertial SLAM Using Recurrent Field Transforms

Wei Zhang[1,2], Sen Wang[2,3], Xingliang Dong[4], Rongwei Guo[4] and Norbert Haala[1]

***Abstract*— In this paper, we present BAMF-SLAM, a novel multi-fisheye visual-inertial SLAM system that utilizes Bundle Adjustment (BA) and recurrent field transforms (RFT) to achieve accurate and robust state estimation in challenging scenarios. First, our system directly operates on raw fisheye images, enabling us to fully exploit the wide Field-of-View (FoV) of fisheye cameras. Second, to overcome the low-texture challenge, we explore the tightly-coupled integration of multi-camera inputs and complementary inertial measurements via a unified factor graph and jointly optimize the poses and dense depth maps. Third, for global consistency, the wide FoV of the fisheye camera allows the system to find more potential loop closures, and powered by the broad convergence basin of RFT, our system can perform very wide baseline loop closing with little overlap. Furthermore, we introduce a semi-pose-graph BA method to avoid the expensive full global BA. By combining relative pose factors with loop closure factors, the global states can be adjusted efficiently with modest memory footprint while maintaining high accuracy. Evaluations on TUM-VI, Hilti-Oxford and Newer College datasets show the superior performance of the proposed system over prior works. In the Hilti SLAM Challenge 2022, our VIO version achieves second place. In a subsequent submission, our complete system, including the global BA backend, outperforms the winning approach.**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is being widely deployed to real-world applications in various domains such as mobile robotics and augmented reality. In the active field of SLAM research, Visual-Inertial SLAM (VI-SLAM) has emerged as a highly robust and accurate solution. Despite significant progress in VI-SLAM research [1], [2], [3], [4], emerging challenging datasets and new industrial applications [5], [6] present new formidable challenges that demand increased accuracy and robustness from state-of-the-art VI-SLAM systems. For example, monitoring construction sites requires centimeter-level accuracy in low-texture environments with rapid lighting changes and motion. To address these challenges, we opt for the use of multiple fisheye cameras in combination with the complementary Inertial Measurement Unit (IMU) as sensor setup. This provides the system with sufficient sensing information, about both the environment and ego motion to handle challenging cases, resulting in more robust and efficient environment mapping.

[1]Institute for Photogrammetry, University of Stuttgart, Germany wei.zhang, norbert.haala@ifp.uni-stuttgart.de
[2]Audiovisual Lab, Huawei Munich Research Center, Germany wei.zhang3, sen.wang@huawei.com
[3]CAMP, Technical University of Munich, Germany sen.wang@tum.com
[4]Central Media Technology Institute, Huawei 2012 Laboratories, China dongxingliang, guorongwei@huawei.com

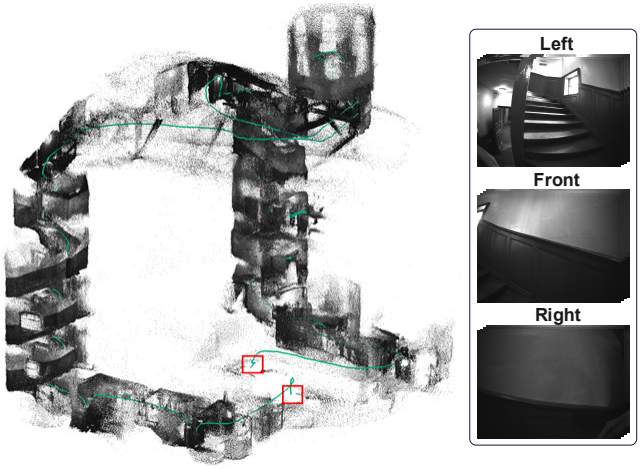

Fig. 1. Estimated point cloud and trajectory by the proposed system on exp09 sequence of Hilti-Oxford dataset. The start and end points of the trajectory are marked with red rectangles.

Bundle adjustment [7] has become the gold standard in modern SLAM systems. In feature-based approaches [3], [4], the objective is to minimize the reprojection errors of the detected sparse features. In contrast, the photometric errors of dense pixels are minimized instead in direct approaches [8], [9]. With the increase of computational power and advances in deep learning, recent works [10], [11], [12] tend to combine the strengths of both worlds, which is to minimize the reprojection errors of dense depth maps using the prediction of RFT as targets. Based on this scheme, we built our system and extended it to a multi-camera VI-SLAM setup, which includes a unified joint factor graph formulation to fuse the inputs of multi cameras and IMU measurements in a tightly-coupled manner.

One major limitation of most visual SLAM systems is its reliance on visual appearance inside the camera view. They struggle to track the camera pose when the camera view is obstructed, contains too little texture, or has low overlap due to rapid movements. The great advantage of a multi-fisheye VI-SLAM system is that, on the one hand, the large FoV of fisheye cameras reduces the likelihood of low textures and increases the view overlap. When one view is fully obstructed, the other cameras or the IMU measurements can still provide redundancy and prevent the system from tracking failure or catastrophic drifting.

For global consistent mapping, we propose an approach called semi-pose-graph BA to jointly optimize poses and the depth maps involved by detected loop closures. Compared to the full BA, our method does not rely on reprojection factors to connect all keyframes, but instead converts the

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160905

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:57:39 UTC from IEEE Xplore. Restrictions apply.

reprojection factors computed by the frontend into relative pose edges to build a pose graph. In contrast to pose graph optimization (PGO), our method optimizes not only the poses but also the depth maps of keyframes involved in detected loop closures, which are represented by reprojection factors. Thus our method is more accurate than PGO while also being more efficient than full BA.

In summary the main contributions of this work are:

- A state-of-the-art multi-fisheye VI-SLAM system called BAMF-SLAM that utilizes full image information from multiple fisheye cameras as well as complementary IMU measurements.
- A unified factor graph formulation that tightly-couples multi-camera inputs and IMU measurements to jointly optimize poses and depth maps.
- A semi-pose-graph bundle adjustment scheme that leverages relative pose edges and loop closures as reprojection factors to achieve efficient loop closing while maximizing global consistency.

II. Related Work

Both visual-inertial SLAM and multi-camera SLAM have received extensive research. To fuse visual and inertial measurements, most works, including this one, use the preintegration method of [13] to reduce the number of variables and efficiently solve the bundle adjustment problem. For a fixed computation cost of frontend, the marginalization of old states [1] has been popular to model their uncertainty, whereas another line of works [3], [14] includes the old states as fixed prior in local BA. This work adopts the latter approach due to its simplicity and effectiveness.

Several works have explored the use of multiple cameras in SLAM systems. [15] proposes a general design for multi-camera SLAM system including an initialization strategy and keyframe selection. MultiCol [16] extends the ORB-SLAM2 [17] for the use with multiple fisheye camera inputs via bundle adjustment. [18] proposes a panoramic camera model for combining the images of a multi-camera rig, but only applicable when the camera centers have a slight offset. Our sensor configuration does not meet the requirement and thus cannot use the panoramic model.

While there have been a few previous attempts to combine multi-camera inputs and inertial measurements, most methods rely on feature-based tracking across cameras. For instance, VILENS-MC [19] leverages cross-camera feature tracking to exploit camera overlaps. In contrast, our system does not track feature points, but instead establishes dense pixel-wise correspondences based on depth map representation. Furthermore, our system includes a global optimization backend resulting in further improvements in accuracy and robustness compared to previous works.

III. Preliminary on Recurrent Field Transforms

The concept of recurrent field transforms can be traced back to RAFT [20], a deep network architecture for optical flow estimation. Inspired by this success, RAFT-3D [21] and RAFT-Stereo [22] adopted this concept for the tasks of scene

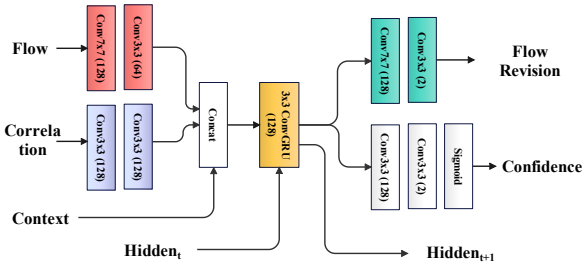

Fig. 2. Network structure for the update operation of recurrent field transforms borrowed from [10].

flow estimation and stereo matching respectively. Later, the concept of RFT was adopted for the dense SLAM system, DROID-SLAM [10], which combines RFT with BA to solve poses and depths from optical flows. RFT predicts the dense pixel-wise correspondences between co-visible views based on the feature maps extracted from the input images using a residual-style network. The feature correlation can be looked up using the optical flow derived from the latest pose and depth estimates. As depicted in Fig. 2, the inputs to the GRU module include, the current correlation, the context features, the last flow update, and the last hidden state. They are concatenated together to compute the update of the next flow revision and the associated confidence values.

To improve RFT-BA cooperation, DROID-SLAM enhances the training process by incorporating BA as a differential layer, allowing the loss function to be directed at pose and depth map estimates rather than optical flow and achieves the state-of-the-art accuracy on multiple benchmarks. This work builds based on DROID-SLAM and extends it to a multi-camera VI-SLAM system to deal with more challenging scenarios encountered in real-world applications.

Despite only being trained on the dataset [23] of pinhole images, the pre-trained model by [10] has excellent generalization ability on fisheye images. Therefore, we use the pre-trained model without further finetuning.

IV. System Overview

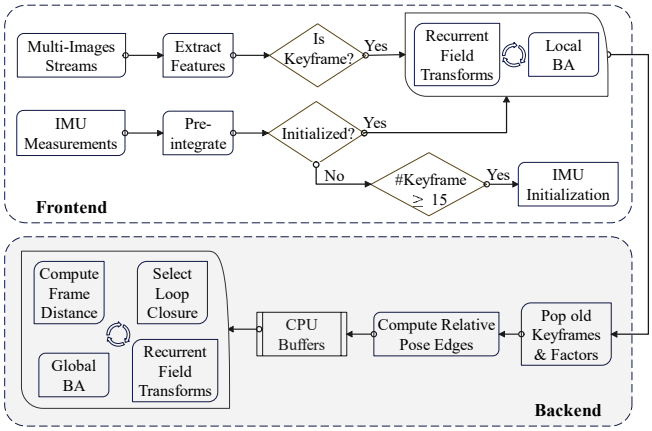

Fig. 3. Overview of the proposed system.

The task of our proposed system is to estimate the state of the sensor platform as accurately and robustly as possible by utilizing the onboard multi-fisheye cameras and an IMU sensor. Figure 3 depicts a high-level overview of our system. Following the design of modern SLAM frameworks, we

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:57:39 UTC from IEEE Xplore. Restrictions apply.

divide our system into frontend and backend. The frontend is responsible for processing the input images and IMU measurements, determining which are keyframes, keeping track of the most recent keyframes in a sliding window, and performing alternative RFT and local BA optimization. On the backend, it manages a data buffer of all keyframes established by the frontend and converts the intermediate results of reprojection factors into relative pose factors to build a global pose graph. Finally, to improve overall accuracy and global consistency, the backend employs loop closure detection and global semi-pose-graph BA.

V. Method

This section describes the proposed system's methodology in detail. We begin by discussing the application of the fisheye camera model to leverage the wide fisheye FoV in Sec. V-A. Sec. V-B covers the crucial IMU initialization process, which enables our system to combine visual-inertial information. Next, we present the factor graph formulation and explain each type of factors that are employed for fusing multiple inputs in Sec. V-C. Finally, we introduce our proposed semi-pose-graph BA method (Sec. V-D) that achieves global consistency and improved accuracy by identifying long-term loop closures and maximizing the agreement between relative pose and loop closure constraints.

A. Fisheye Camera Model

Conventional SLAM methods often assume an ideal pinhole camera model, which requires pre-processing of input images to remove distortion. However, when it comes to fisheye images, this can result in a loss of useful information due to cropped borders or sampling artifacts caused by the interpolation of objects at the edges. This negates the benefits of the wide FoV that fisheye images offer. To address this issue, our proposed system uses the Kannala-Brandt fisheye camera model [24] as in [3] throughout our framework. The model is used for image pixel projection and unprojection, as well as the analytical computation of the Jacobians in the (uni-)projection functions. Unlike [3], we use depth maps instead of sparse features and implement a custom CUDA kernel to parallelize the operation for all pixels.

B. IMU Initialization

To enable the smooth integration of IMU observations, a set of new states that are not estimated in vision-only system need to be initialized, including gravity direction, scale, body velocity and biases of gyroscope and accelerometer. Following the best practice in [3], it is preferable to start with vision-only BA for a short sequence with sufficient motion before initializing IMU. To summarize, our IMU initialization procedure consists of the following four steps:

1) Vision-only BA to establish the initial poses of the first K keyframes.
2) (Optional for static start) Least-square optimization for an initial gravity direction R_{wg} and accelerometer bias b_a using the constraint $\bar{g} - b_a = R_{wg}(0, 0, G)^T$, where G is the gravity magnitude and $\bar{g}$ is obtained by averaging the accelerometer measurements at standstill.
3) Inertial-only BA to jointly optimize velocity, scale, biases b_g as well as refine the estimates of the last two steps, namely R_{wg} and b_a using the preintegrated IMU factors (See Sec. V-C).
4) Visual-Inertial BA to finally optimize all variables using both visual reprojection and IMU preintegration factors.

In step 2, we determine whether the initial phase is stationary by examining the magnitude of the gyroscope measurement. If the initial phase is indeed stationary, the accelerometer data in this phase can provide strong prior information about the gravity direction.

The scale can typically be observed directly through a known stereo baseline via offline calibration and does not require optimization as a variable. However, our empirical study found that multi-camera calibration often lacks accuracy, and the inclusion of a scale variable can reduce the impact of calibration errors.

C. Factor Graph Formulation

The core to our bundle adjustment based system lies in the formulation of a factor graph that incorporates all types of factors to optimize the system states and 3D structure jointly. The 3D structure is represented by the depth maps of keyframes. The system state is defined in the IMU frame denoted by B and the state at time t_i can be defined as follows:

$$\mathbf{x_i} = [\mathbf{T_i}, \mathbf{v_i}, \mathbf{b_i^a}, \mathbf{b_i^g}]^T \tag{1}$$

where $\mathbf{T_i} = [\mathbf{R}_i, \mathbf{p}_i] \in$ SE(3) is the body pose, $\mathbf{v_i^w}$ is the velocity, and $\mathbf{b_i^a}$, $\mathbf{b_i^g}$ are the IMU accelerometer and gyroscope biases respectively.

Fig. 4 depicts the factor graph formulations for the frontend and backend respectively. On the frontend, we maintain a sliding graph and optimize only the variables within a dynamic window. Upon a new keyframe, we first optimize the state of the incoming keyframe and its depth map, using a window size of 1, which we refer to as the tracking process. Next, the local BA is performed with a larger window size of n keyframes. Through empirical observation, we have observed that directly running local BA on new keyframe requires more iterations to converge. However, if we perform local BA after tracking, which is significantly faster than local BA, it only takes two iterations to converge. This approach results in a significant improvement in runtime performance, as discussed in Sec. VI-E. Moreover, a few vertices outside the window linked by reprojection factors are included in the optimization as fixed priors.

To incorporate IMU measurements, we utilize the IMU preintegration method [13] to compute IMU preintegration factors that connect consecutive keyframes. For reprojection factors, we distinguish between the mono and stereo types. Mono factors connect a pair of keyframes with common view, where vertex pose estimates are used to transform the coordinates during reprojection. In the stereo case, cross-camera reprojection is computed using the known camera extrinsics. Let Π denote the fisheye projection function, a

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:57:39 UTC from IEEE Xplore. Restrictions apply.

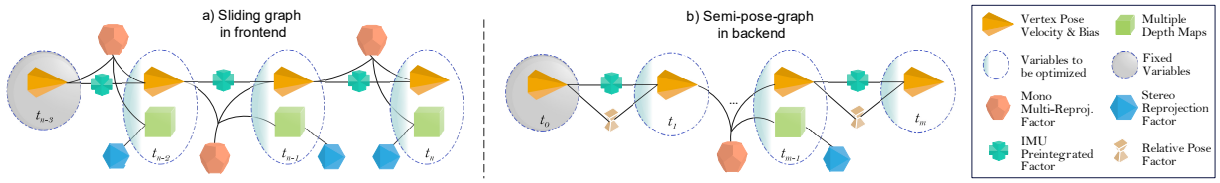

Fig. 4. Illustration of the proposed joint factor graph formulation. a) A sliding graph with a window size of three keyframes and one fixed keyframe served as prior. b) Semi-pose-graph with a long-term loop closure between two distant keyframes shown as an example.

point $\mathbf{p}_i$ can be reprojected with an estimated depth $\hat{d}_i$ using the following equation:

$$\hat{\mathbf{p}}_{ij} = \Pi(\hat{\mathbf{T}}_{ij}\Pi^{-1}(\mathbf{p}_i, \hat{d}_i)) \tag{2}$$

where $\hat{\mathbf{T}}_{ij} = T_{CB}\hat{\mathbf{T}}_j^{-1}\hat{\mathbf{T}}_i T_{CB}^{-1}$ refers to the relative transformation from keyframe i to keyframe j in the mono case. In the stereo case, it refers to the extrinsic $T_{C_iC_j}$ between camera i and camera j.

For the reprojection targets, the RFT predicts the flow revision $\boldsymbol{\delta}_{ij}$ to add on the last optical flow $\hat{\mathbf{p}}_{ij}$. Thus, the updated optical flow, namely the dense correspondences, is obtained as follows:

$$\check{\mathbf{p}}_{ij} = \hat{\mathbf{p}}_{ij} + \boldsymbol{\delta}_{ij} \tag{3}$$

To mitigate the impact of false correspondence and occlusion, the confidences w_i is predicted for each pixel by RFT. Low confidences are assigned to pixels that do not have a counterpart in the target view or are ambiguous. The reprojection error then can be formulized as follows:

$$r_{ij} = \|\check{\mathbf{p}}_{ij} - \Pi(\hat{\mathbf{T}}_{ij}\Pi^{-1}(\mathbf{p}_i, \hat{d}_i))\|^2_{w_i} \tag{4}$$

As a special case in our multi-camera setup, one keyframe has multiple depth maps pointing to different view directions. In this case, the reprojection errors are computed for all depth maps, which we refer to as the multi-reprojection factor. For Newer College and Hilti-Oxford dataset, we include the depth maps of the left and right views in addition to the front view.

D. Semi-Pose-Graph BA

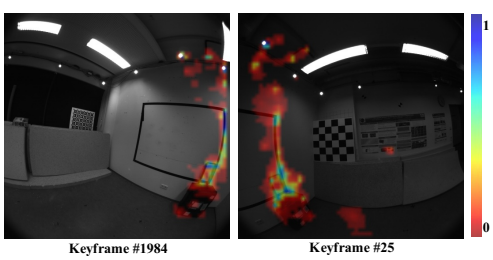

Fig. 5. Example of a wide baseline loop closure with color-coded confidence values predicted by the RFT network. Note that values lower than 0.01 are omitted for clarity.

Performing global full BA is computationally expensive and requires a large memory footprint as it optimizes the system states and depth maps of all keyframes using all reprojection factors by the frontend. Additionally, since the frontend already provides states with high relative accuracy and local consistency, it may not be necessary to optimize the already very accurate relative pose. To this end, we have determined that the frontend's reprojection factors can be converted into relative pose factors using the dense correspondences established by local BA. A pose graph can then be constructed using these relative pose factors. The geometry constraints underlying the reprojection factors is primarily represented by the relative pose factors and their associated covariances. We compute these relative pose factors and covariances using the Gauss-Newton method [25] as follows:

$$\mathbf{H}\boldsymbol{\Delta}\mathbf{x} = \mathbf{J}^T\mathbf{W}\mathbf{J}\boldsymbol{\Delta}\mathbf{x} = \mathbf{b} \tag{5}$$

$$\mathbf{Cov} = (\mathbf{J}\boldsymbol{\Delta}\mathbf{x} - \mathbf{r})^T\mathbf{W}(\mathbf{J}\boldsymbol{\Delta}\mathbf{x} - \mathbf{r})\mathbf{H}^{-1} \tag{6}$$

where $\mathbf{J}$ is the Jacobian of the reprojection function w.r.t the relative pose, $\mathbf{r}$ is the last reprojection error vector and $\mathbf{W}$ is the weight matrix composed by the confidence predicted by RFT. The update to the relative pose is denoted as $\boldsymbol{\Delta}\mathbf{x}$.

The trajectory estimate by the frontend is prone to odometry drift, which accumulates over long distances traveled. To mitigate drift error, the backend searches for potential loop closures to add as additional constraints. The backend starts by computing the frame distances between all keyframes, which is determined by the mean optical flow calculated based on the last pose and depth map estimates. A relaxed frame distance threshold is typically used to allow for more potential loop closures with wider baselines. While this may lead to the inclusion of wrong loop closures, their impact is insignificant because the RFT network is highly reliable and assigns low confidence to the wrong matches. Fig. 5 depicts a wide baseline loop closure, where the same room is revisited after a long trip. The wide FoV of the fisheye camera used in our system enables the detection of such loop closures even with little overlap at the image border.

With the use of relative pose factors, a pose graph can be constructed. In addition, the detected loop closures can be presented in the form of reprojection factors, which extend the pose graph into a semi-pose-graph that includes depth maps involved in loop closures, as shown in Fig. 4,b). The semi-pose-graph BA is more efficient than full BA, as the number of variables to be optimized is much smaller. Moreover, the relative pose factors, which are derived from the reprojection factors established by the frontend, are highly accurate and robust due to the significant overlap of keyframes within the frontend sliding window. As a result, the semi-pose-graph BA is both efficient and accurate.

VI. Experiments

Extensive evaluations has been carried out on our proposed system using diverse datasets. To validate the effectiveness of our system in stereo+IMU setup, we evaluate in various indoor scenes from TUM-VI dataset [5]. We further evaluate the multi-camera setup on the challenging Newer College dataset [26], which contains indoor and large-scale outdoor

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:57:39 UTC from IEEE Xplore. Restrictions apply.

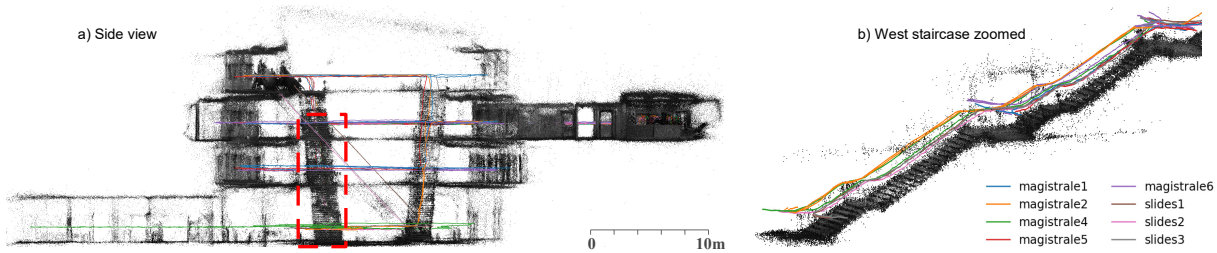

Fig. 6. Qualitative results on TUM-VI dataset. As groundtruth only covers a single room, we transform all results to the groundtruth frame for consistency check. Estimated trajectories are depicted in colors and point clouds are accumulated. Red-box marked region is zoomed in on the right from the side look to provide a better illustration. Notably our results demonstrate high consistency in cleanly separating floors and clear-cut edges of stairs.

scenes, and the Hilti-Oxford dataset [6], which includes challenging sequences collected on construction sites and a historical theatre. For all evaluations, both VIO and VI-SLAM results are provided to differentiate the performance in terms of odometry drift by frontend and overall performance with global optimization. Finally, we present an analysis of the processing time and demonstrate the potential of real-time capability.

A. Implementation Details

The framework is implemented in Python/CUDA using PyTorch and Eigen libraries, and tested on a desktop PC with an Intel i9-12900K CPU and an Nvidia RTX 3090Ti GPU. Input images are downsampled to 440×440 and 384×512 resolution for TUM-VI and the other two datasets respectively. To reduce the memory footprint, a ring buffer data structure is implemented for storing image features and intermediate results of keyframes. This way, only a fixed amount of GPU memory needs to be allocated regardless of data length, and the data of each new incoming keyframe is written to the beginning of the buffer. Our system can run on GPUs with 11GB of memory for both the frontend and the backend, which ensures high performance even for large-scale datasets.

B. Performance on TUM-VI Dataset

The TUM-VI [5] dataset was collected using a handheld device equipped with two fisheye cameras in a stereo setup and an IMU. We selected four challenging scenes, including low-texture (slides), rapid rotation (corridor), and large-scale multi-story building (magistrale), to evaluate our approach. Tab. I summarizes the quantitative evaluation results and the comparison to the previous works. We compared our VIO setup against BASALT [27] and the VIO version of OKVIS2 [4], both of which are feature-based VIO methods without loop closure. Our method outperforms all other methods in terms of averaged ATE across all four scenes.

For the evaluation on VI-SLAM systems, the trajectory optimized globally with loop closures is evaluated. Our full SLAM system outperforms both OKVIS2 [4] and ORB-SLAM3[3], achieving the lowest ATE. Notably, our system's ability to detect and close loop closures with little overlap allows us to reach an ATE of 1cm for almost all sequences. Fig. 5 shows a wide baseline loop closure from the sequence *corridor 1*. On the sequence *magistrale 3*, due to the larger odometry drift and thus a greater frame distance, loop closures cannot be successfully found.

TABLE I

ABSOLUTE TRAJECTORY ERROR (ATE) IN METERS ON THE TUM-VI DATASET [5]. BEST RESULTS ARE IN BOLD.

		Stereo-VIO			Stereo-VI-SLAM			
	Seq	BASALT	OKVIS2	Ours	ORB-SLAM3	OKVIS2	Ours	Length [m]
corridor	1	0.34	0.54	**0.31**	0.03	0.02	**0.01**	305
	2	0.42	0.44	**0.15**	0.02	0.06	**0.01**	322
	3	0.35	0.55	**0.21**	0.02	0.03	**0.01**	300
	4	0.21	0.11	**0.04**	0.21	0.10	**0.01**	114
	5	0.37	0.51	**0.17**	**0.01**	0.09	**0.01**	270
	avg	0.34	0.43	**0.18**	0.06	0.06	**0.01**	262
magistrale	1	**1.20**	2.03	1.74	0.24	0.07	**0.01**	918
	2	1.11	3.22	**1.00**	0.52	1.22	**0.01**	561
	3	**0.74**	2.11	1.53	1.86	**0.09**	1.39	566
	4	1.58	1.94	**1.26**	0.16	0.25	**0.01**	688
	5	0.60	1.01	**0.21**	1.13	0.02	**0.01**	458
	6	3.23	2.32	**1.66**	0.97	0.76	**0.01**	771
	avg	1.41	2.10	**1.23**	0.81	0.40	**0.24**	660
room	1	0.09	0.06	**0.02**	**0.01**	**0.01**	**0.01**	146
	2	**0.07**	**0.07**	**0.07**	**0.01**	**0.01**	**0.01**	142
	3	0.13	0.06	**0.04**	**0.01**	**0.01**	**0.01**	135
	4	0.05	**0.02**	**0.02**	**0.01**	**0.01**	**0.01**	68
	5	0.13	**0.02**	**0.02**	**0.01**	**0.01**	**0.01**	131
	6	0.02	**0.02**	**0.02**	**0.01**	**0.01**	**0.01**	67
	avg	0.08	0.04	**0.03**	**0.01**	**0.01**	**0.01**	115
slides	1	**0.32**	0.96	0.43	0.41	0.37	**0.01**	289
	2	**0.32**	0.74	0.51	0.49	0.16	**0.01**	299
	3	0.89	2.51	**0.41**	0.47	0.13	**0.01**	383
	avg	0.51	1.40	**0.45**	0.45	0.22	**0.01**	324

Notably, the groundtruth is only available at the beginning and end of the sequences, limiting the groundtruth coverage to motion within a single room. Therefore, the ATE reported in Tab. I do not fully represent the accuracy of the estimated trajectory outside of the room. To qualitatively evaluate the overall accuracy, we transform the results to the groundtruth frame using SE3 alignment and examine the consistency of the accumulated point clouds. Fig. 6 shows the accumulated point clouds along with their respective trajectories, depicted in different color.

C. Performance on Newer College Dataset

TABLE II

EVALUATIONS ON NEWER COLLEGE DATASET BASED ON RELATIVE POSE ERROR (RPE) AND ATE (UNIT: METER).

	10m RPE				ATE				
					Our VIO		VI-SLAM		
Seq	Open VINS	ORB-SLAM3	VILENS MC	Ours MC-VIO	Stereo	MC	Stereo	MC	Length [m]
MATH	0.65	Fail	0.26	**0.12**	0.47	0.40	0.21	**0.20**	329
MINE	0.98	Fail	0.20	**0.07**	0.28	0.45	**0.08**	0.11	236
QUAD	1.01	0.23	0.31	**0.18**	0.73	0.64	**0.24**	0.37	244
STAIRS	0.33	0.20	0.16	**0.09**	0.09	0.07	0.05	**0.04**	59
PARK	0.18	0.13	N/A	**0.10**	4.27	5.08	**0.53**	1.04	2396

The Newer College dataset [26] includes four fisheye cameras and an IMU sensor. The difficulty and variety of

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:57:39 UTC from IEEE Xplore. Restrictions apply.

the scenes range from narrow staircase, to dark underground mine, and over 2 km outdoor walk. The groundtruth is provided for the entire sequence by aligning lidar scans to detailed prior maps. Tab. II presents the evaluation results on the Newer College dataset. Following [19], we compare our VIO system to OpenVINS [28], ORB-SLAM3 [3] and VILENS-MC [19] based on the metric Relative Pose Error (RPE) over 10m. Our results demonstrate that our method outperforms all other methods on all scenarios. Notably, VILENS-MC, which also uses a multi-camera setup, is the closest method to our approach, highlighting the benefits of leveraging multi-camera inputs.

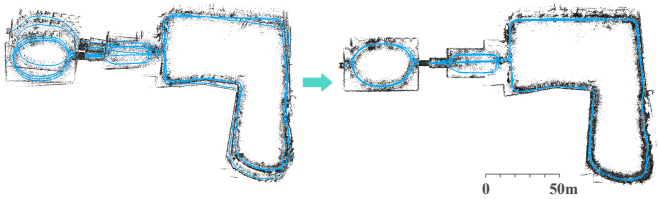

Fig. 7. Before and after global semi-pose-graph BA on sequence *Park*.

In addition to using relative metrics, we also assess absolute accuracy using the ATE metric. Surprisingly, our experiments show that the error in the stereo setting is lower than in the multi-camera (MC) setting. We suspect this discrepancy may be due to errors in the extrinsic calibration and plan to investigate further in future work. When comparing the VIO and VI-SLAM results, the VI-SLAM results are significantly better than the VIO results thanks to the loop closing and semi-pose-graph BA. It could be argued that the VIO frontend has provided a fair good starting point for the backend, with relatively low odometry drift. Based on this, the backend was then able to identify potential loop closures by measuring frame distances between keyframes.

Fig. 7 illustrates the effectiveness of our backend, showing a significant improvement before and after the semi-pose-graph BA on the long walking sequence *Park*. Over 2 km of walking and revisits through parks and corridors, the estimated trajectory remains smooth and highly consistent, as evidenced by the clean building edges and park walls in the point cloud map.

D. Performance on Hilti-Oxford Dataset

TABLE III

ATE IN METER AND OBTAINED SCORES ON HILTI-OXFORD DATASET.

	exp01	exp02	exp03	exp07	exp09	exp11	exp15	exp21	score
OKVIS2	-	-	-	-	-	-	-	-	32.5
Our VIO	0.16	0.28	0.30	0.15	0.38	0.17	0.18	0.30	22.2
Our SLAM	0.10	0.15	0.15	0.14	0.28	0.21	0.17	0.30	40.9

The Hilti-Oxford dataset [6] is part of the Hilti SLAM challenge 2022, and it features a sensor configuration similar to that of the Newer College dataset, consisting of five fisheye cameras and an IMU. To reduce the computational resources needed, we chose four views from five cameras for processing, omitting the top view as it has fewer features. The dataset provides millimeter-accurate groundtruth at sparse locations and a web interface for evaluation as well as a score leaderboard are provided for benchmarking. The RMSE ATE on sparse groundtruth is used as the evaluation metric, and the results are summarized in Tab. III. Our VIO system achieved ATEs ranging from 10 cm to 30 cm. Exp01-exp03 were collected in a construction site with start and end points at the same place, allowing for loop closures. Thus our full-SLAM system shows improvements for these sequences, resulting in a higher score that outperforms the winner approach of Hilti SLAM challenge 2022 by OKVIS2.

E. Processing Time

TABLE IV

PROCESSING TIME OF DIFFERENT MODULES (UNIT: MILLISECOND).

	IO	Extract eature	Check keyframe	Preintegrate IMU	Tracking	Local-BA	Compute rel. pose	Total
Non-keyframe	7	4	2	-	-	-	-	13
Keyframe	7	4	2	3	48	106	8	178

The runtime performance of the system is evaluated on the TUM-VI dataset for both non-keyframe and keyframe cases. The results are presented in Tab. IV, which breaks down the time into different stages. For non-keyframes, the incoming image is processed up to the keyframe-check step. The average time taken for non-keyframes is 13 milliseconds. In contrast, each keyframe is further tracked by running four iterations single-frame BA followed by two iterations of local BA. Finally, the relative pose factors are computed for the keyframes and their associated reprojection factors outside current sliding window. On average, it takes around 175 milliseconds to process one keyframe.

The system achieves a good balance between processing speed and accuracy, selecting on average 23.5% of the frames as keyframes while still processing an average of 25 frames per second on the TUM-VI dataset, exceeding the recording time by 25%. However, in multi-camera mode, the increased computation reduces the processing speed to an average of 22 frames per second on the Newer College and Hilti-Oxford dataset. On the backend, loop closuring and global semi-pose-graph BA are carried out once by the end of program, which can take several seconds to a few minutes for long sequences such as the Newer College *Park* sequence.

VII. CONCLUSION

This paper presents a state-of-the-art visual-inertial SLAM system capable of fusing visual reprojection factors from multiple fisheye cameras and IMU preintegration factors into a unified factor graph. To achieve globally consistent mapping, a novel semi-pose-graph bundle adjustment scheme has been proposed to replace the computationally expensive full BA. By running alternative BA optimizations and RFT predictions, the system can converge to the maximum likelihood solution. Extensive evaluations on multiple datasets validate the accuracy and robustness of our proposed system. As part of future work, we aim to enhance the robustness in large-scale outdoor scenes, where noisy pixels like sky and shadows should be better handled. Furthermore, we plan to incorporate a global loop closure detection method, such as Bag-of-Word, to strength the loop closing ability in case of large odometry drift caused by long-distance travel.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:57:39 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

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

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[4] S. Leutenegger, "OKVIS2: Realtime scalable visual-inertial slam with loop closure," *arXiv preprint arXiv:2202.09199*, 2022.

[5] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1680–1687.

[6] L. Zhang, M. Helmberger, L. F. T. Fu, D. Wisth, M. Camurri, D. Scaramuzza, and M. Fallon, "Hilti-oxford dataset: A millimeter-accurate benchmark for simultaneous localization and mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 408–415, 2022.

[7] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*. Springer, 2000.

[8] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[9] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[10] Z. Teed and J. Deng, "DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 558–16 569, 2021.

[11] W. Zhang, S. Wang, and N. Haala, "Towards robust indoor visual SLAM and dense reconstruction for mobile robots," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, pp. 211–219, 2022.

[12] P. Ji, Q. Yan, Y. Ma, and Y. Xu, "Georefine: Self-supervised online depth refinement for accurate dense mapping," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*. Springer, 2022, pp. 360–377.

[13] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation." Georgia Institute of Technology, 2015.

[14] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[15] J. Kuo, M. Muglikar, Z. Zhang, and D. Scaramuzza, "Redesigning SLAM for arbitrary multi-camera systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2116–2122.

[16] S. Urban, S. Wursthorn, J. Leitloff, and S. Hinz, "MultiCol bundle adjustment: a generic method for pose estimation, simultaneous self-calibration and reconstruction for arbitrary multi-camera systems," *International Journal of Computer Vision*, vol. 121, no. 2, pp. 234–252, 2017.

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

[18] S. Ji, Z. Qin, J. Shan, and M. Lu, "Panoramic SLAM from a multiple fisheye camera rig," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 169–183, 2020.

[19] L. Zhang, D. Wisth, M. Camurri, and M. Fallon, "Balancing the budget: Feature selection and tracking for multi-camera visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1182–1189, 2021.

[20] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *European conference on computer vision*. Springer, 2020, pp. 402–419.

[21] Z. Teed and J. Deng, "RAFT-3D: Scene flow using rigid-motion embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8375–8384.

[22] L. Lipson, Z. Teed, and J. Deng, "RAFT-Stereo: Multilevel recurrent field transforms for stereo matching," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 218–227.

[23] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "TartanAir: A dataset to push the limits of visual SLAM," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4909–4916.

[24] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[25] W. Niemeier, "Ausgleichungsrechnung," in *Ausgleichungsrechnung*. de Gruyter, 2008.

[26] L. Zhang, M. Camurri, and M. Fallon, "Multi-camera LiDAR inertial extension to the newer college dataset," *arXiv preprint arXiv:2112.08854*, 2021.

[27] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 422–429, 2019.

[28] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:57:39 UTC from IEEE Xplore. Restrictions apply.