

# Rumination Meets VSLAM: You Don't Need to Build All the Submaps in Realtime\*

Weinan Chen, Changfei Fu, Lei Zhu, Shing-Yan Loo, Hong Zhang\* IEEE Fellow

**Abstract**—In the application of visual navigation, submap-based VSLAM has become one of the most robust monocular solutions in recent years, which is able to resume tracking by multi-submap maintenance and merging. However, due to the lack of long-term data association between submaps, global consistency cannot be guaranteed in the existing work, especially in situations without loop-closure. Considering the fact that not all the submap have to be built in realtime, we propose a VSLAM system with realtime and non-realtime hybrid style, RUMI-SLAM. Inspired by the rumination of mammals that processes food in various stomachs and absorbs it in one stomach, RUMI-SLAM performs asynchronous submap building and centralized submap management. Building additional submaps in parallel leads to enriched mapping elements and enhanced data association across submaps. The experimental results demonstrate the superiority of RUMI-SLAM over the existing VSLAM systems, especially the robustness to challenging situations. We also provide real-robot experiments to demonstrate our RUMI-SLAM in the application of visual navigation. Our study provides a novel asynchronous submap-based VSLAM framework, which achieves globally consistent submap merging without the requirement of loop-closure.

**Index Terms**—VSLAM, Submap Merging, Rumination

## I. INTRODUCTION

As the core of visual navigation, monocular VSLAM (Visual Simultaneous Localization And Mapping) has been extensively studied in recent years [1]). Among the existing popular VSLAM systems, they are built upon a single map, the system cannot maintain running after tracking loss and mapping interruption [2]. The development of submap-based VSLAM has improved the tracking robustness by a multi-submap representation [3], where each submap is part of a global map [4], including the elements such as frames and map points. In challenging situations, such as motion blur and texture-less scenes, visual tracking can be resumed by creating a new submap. Due to the discontinuous observation and multiple submaps, submap merging determines the global consistency of the environment representation. Essentially, the

\* The work in this paper is supported by the National Natural Science Foundation of China (62103179).

Weinan Chen and Lei Zhu are with the Biomimetic and Intelligent Robotics Lab, Guangdong University of Technology, Guangzhou, China.

Hong Zhang and Changfei Fu are with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China.

Shing-Yan Loo is with Computing Science Department, University of Alberta, Canada.

Weinan Chen and Changfei Fu contribute equally to this paper.

Manuscript received April –, –; revised August –, –.

submap merging is achieved by long-term data association introduced by loop-closure, which matches current observations with elements in previously visited areas.

As listed in Tab. I, three types of data association are included in submap-based VSLAM systems [5]:

**Short-term data association** is the matching between current observation and elements in the last few seconds, which is in realtime for visual tracking.

**Mid-term data association** is the matching between current observation and elements nearby, which is for local bundle adjustment (BA) over frames and their map points.

**Long-term data association** is the matching between current observation and previously visited elements for submap merging and loop correction [6]. It is worth noting that long-term data association is not required to be in realtime, since the realtime tracking and local mapping are achieved by the short-term and mid-term data association.

In the application of VSLAM using a camera equipped in a mobile robot [7], we focus on enhancing long-term data association for robust submap merging [8]. Submap-based VSLAM has achieved robust performance to a certain extent, while the long-term data association across submaps cannot be guaranteed due to the untracked challenging frames and the assumption of loop-closure. Therefore, the submap merging without the requirement of loop-closure, or the enhancement of long-term data association is still an open problem. To solve this problem, we propose an asynchronous submap-based VSLAM system, which has a rumination-inspired framework, called RUMI-SLAM. We propose to enhance long-term data association and global consistency using a combination of realtime processing and non-realtime processing, without the requirement of loop-closure trajectory, as shown in Fig. 1. The suboptimality of non-realtime submap building is an acceptable tradeoff for achieving a globally consistent map. Having the flexibility of building additional submaps in non-realtime fashion provides more opportunities for better long-term data association and global consistency.

The contributions of our study are threefold:

1. We propose an asynchronous submap-based VSLAM system, a biomimetic approach inspired by rumination;
2. The proposed system promotes submap merging without the requirement of loop-closure;
3. We make our system and the evaluation tools code publicly available [9].

TABLE I  
THREE TYPES OF DATA ASSOCIATION IN VSLAM SYSTEMS

	Activation	Considered Element	Function	Submap Operation
Short-term data association	Time	Elements in the last few seconds	Realtime tracking, local BA	Within a submap
Mid-term data association	Location	Elements in nearby	Realtime tracking and mapping, local BA	Within a submap
Long-term data association	Appearance	Elements in visited area	Submap merging and loop correction, global BA	Inter-submap

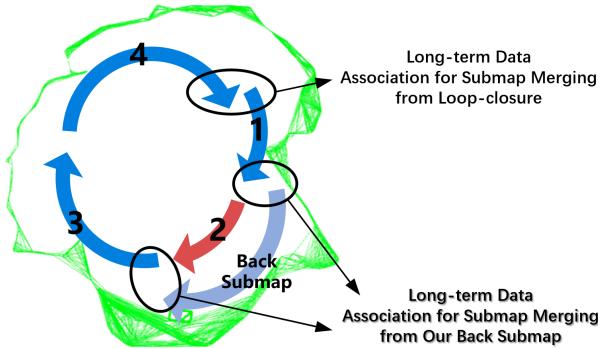


Fig. 1. Comparison with the existing submap-based system. 1-2-3-4 are the sequence parts, part 2 is the sequence failing to track, and the green line is a global map. Visual tracking in submap on part 1 fails due to the challenging frames in part 2, and a new submap on part 3 and 4 is built. For submap merging, a loop-closure (part 4) is required. In RUMI-SLAM, a back submap (light blue arrow) is built parallel to the realtime submap building, establishing the long-term data association without loop-closure.

## II. RELATED WORK

Our proposed work intersects the topics of submap-based VSLAM, image sequence processing and rumination behavior.

### A. Submap-based VSLAM

For reliable performance, a robust graph representation contributes to maintaining visual tracking and mapping. One solution is using a multi-submap representation [10] [11]. Besides the multi-agent mapping [12], the robustness improvement using multi-submap is obtaining attention [13]. To improve robustness in visual tracking, a new submap is initialized in the event of tracking loss to maintain local mapping [14]. The submaps are matched based on the visual similarity in the map intersection. Then, the submaps are merged to form a globally consistent mapping. The previous submap-based VSLAM systems are synchronous [15], in which a new submap is initialized only when the previous one fails to track, and all the submaps are built in realtime. In most cases, the visual tracking failure is caused by challenging frames, which are hard to track. Even though a new submap is built to maintain tracking after tracking failure, the problem is that the inter-submap data association still cannot be produced due to un-tracked challenging frames and their lacked elements. Therefore, loop-closure is required for submap merging, which supplements the elements in un-tracked frames.

As mentioned above, only a part of the elements are considered in the realtime submap building [16], which is efficient but not robust. To achieve robustness to challenging frames, taking time to consider more elements for data association

production is a solution. However, due to the synchronous design of the existing systems [17], the time-consuming operation cannot be realized under the realtime constraint. Furthermore, the assumption of loop-closure trajectory cannot be ensured in practice. To solve this problem, we propose an asynchronous framework, which allows a complementary non-realtime process for building submaps on the challenging frames. Leveraging the fact that long-term data association is not required in realtime, the submap merging and globally consistent mapping are achieved by the additional submaps without the requirement of loop-closure.

### B. Image Sequence Processing in VSLAM

In the aspect of asynchronous image sequence processing, only a few existing studies focus on this problem. In the studies of popular VSLAM systems, such as ORB-SLAM3 [18], SVO [19] and LDSO [20], all of them process images following the time sequence with one process, and they can be summarized as Single Process and Single-Submap (SPSS) systems or Single Process and Multi-Submap (SPMS) systems. In these systems, both the visual tracking and mapping are conducted with realtime input, and only one VSLAM process is executed. Even in the SPMS systems, a new submap is launched only when tracking is lost, which is still a synchronous design with a single process [21].

As for the non-sequential processing, RTAB-Map [22] is a popular system, which performs asynchronous memory management for place recognition. The image is matched across sequences within different memory for efficient retrieval. Compared with RTAB-Map, our system is a total asynchronous system, instead of the place recognition module only. We focus on the monocular configuration, instead of the multi-sensor asynchronous fusion [23]. Compared to the mentioned SPSS and SPMS systems, ours is a **Multi-Process and Multi-Submap (MPMS)** system involving a realtime front-process and a non-realtime back-process. The MPMS design makes use of the distributed submap building mostly, which builds multiple submaps in various processes asynchronously, and the processes are parallel to each other. To our knowledge, the most related work to ours is [24], in which two processes are included for submaps building. However, only one submap is maintained in a single process. Inspired by the rumination behavior, we propose to track images in realtime and non-realtime hybrid style, which satisfies both the realtime requirement by the front-process and the robustness requirement by the back-process.

### C. Rumination

Our proposed system is a MPMS system inspired by rumination. Ruminants share the characteristics that they regur-

gitate and re-chew their feed [25]. The rumination enables ruminants to eat or digest distributed, as is illustrated in 2. Ruminants eat quickly with minimal chewing. After being swallowed, feed enters the reticulum and the rumen. As the largest one of the four stomach, the rumen chambers and serves as a large mixing vat that is the site of microbial fermentation and nutrient absorption. During rumination, the partially digested feed is regurgitated, re-chewed, and re-swallowed. The abomasum produces acid and digestive enzymes similar to the stomach of non-ruminant animals, further breaking down the feed before it passes into the lower gastrointestinal tract for further digestion, absorption and ultimately elimination [26].

In the area of robotics and computer vision, rumination has been studied [27]. To our knowledge, this work is the first study intersecting rumination and VSLAM system framework. We adopt a similar in processing submaps and enhancing long-term data association. Like peristalsis, we process the challenging frames in non-realtime to provide additional submaps for enhancing long-term data association; like absorption, we build a globally consistent map by submap merging and submap management.

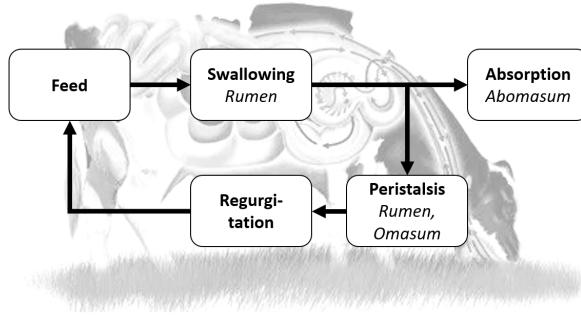


Fig. 2. The illustration of rumination. Five processes are shown: feed, swallowing, peristalsis, regurgitation, and absorption. Food is pre-digested and digested distributed and absorbed centralized.

### III. PROBLEM STATEMENT

#### A. Data Association

Two modules are included in a submap-based VSLAM system, the submap building and the submap merging. In the submap building, given a query frame  $F_q$  with feature  $p_k^q$  and an initialized submap  $M_i = \{P_0^i, \dots, P_{pi}^i, F_0^i, \dots, F_{fi}^i\}$  containing frame  $F_m^i$  and map point  $P_n^i$  (the superscript represents the belonging submap, and the subscript represents the index), the data association between  $F_q$  and  $M_i$  is shown in Formula 1,

$$\text{match}_q = \{p_k^q \mid D(d(p_k^q), d(P_n^i)) < Th_d\} \quad (1)$$

where  $\text{match}_q$  is the matched features in  $F_q$ ,  $d()$  is the description of a map point or a feature,  $D()$  is the distance between two descriptors, and  $Th_d$  is a threshold.  $F_q$  is classified as **untracked**, or **tracking failure**, when the size of  $\text{match}_q$  is less than a given number. VSLAM systems suffer from tracking loss under the influence of untracked frames. For an efficient searching of  $\text{match}_q$  and satisfying the

realtime requirement, only a part of  $P_n^i$  in  $M_i$  is considered, as discussed in Tab. I. For realtime tracking, only elements in the last few seconds are considered, known as short-term data association; to build a local map, only nearby elements are considered, named mid-term data association; for loop-closure detection and submap merging, all the visited elements are considered, known as long-term data association.

To merge the independent submaps, long-term data association across submaps is needed. Given two independent submaps  $M_i$  and  $M_j$ , the data association  $M_{com}$  between them is defined as Formula 2, where  $P_n^i$  is the map point in  $M_i$ , and  $P_t^j$  is the map point in  $M_j$ .

$$M_{com} = \{P_n^i \mid D(d(P_n^i), d(P_t^j)) < Th_d\} \quad (2)$$

The aim of our study is to enhance the long-term data association  $M_{com}$  for submap merging. On the one hand, since  $M_{com}$  is a subset of existing submaps, enriching elements in submaps raises the probability of intersection detection. On the other hand, since the detection of  $M_{com}$  does not have to be in realtime, having additional submaps built non-realtime enhances the long-term data association.

#### B. Sequence Process

Besides the data association between submaps and frames, the sequence processing is formulated. Generally, VSLAM is formulated as  $p(x_{0:t}, M_g | F_{0:t})$  [28], which estimates the robot state  $x$  and the global map  $M_g$  given the observation  $F$ . In a monocular VSLAM system,  $F$  is the captured frame.

As shown in Fig. 1, in existing studies,  $p(x_{0:t}, M_g | F_{0:t})$  is estimated sequentially, which handles the frame sequence obeying the time  $t$ . The estimation is shown in Formula 3:

$$p(x_{0:t}, M_g | F_{0:t}) = \alpha \cdot p(F_t | x_t, M_g) \cdot \sum_t [p(x_t | x_{t-1}) \cdot p(x_{0:t-1}, M_g | F_{0:t-1})] \quad (3)$$

where  $\alpha$  is a constant. As is formulated, to estimate the robot state and the map at time  $t$ , the result in the last moment  $t - 1$  is required for iteration. Therefore, the observation is used sequentially. Based on the formulation above, the existing submap-based VSLAM systems build submaps synchronously within one process.

### IV. METHODOLOGY

To improve the global consistency between submaps, we propose a rumination-inspired MPMS system, named RUMI-SLAM, where two classes of submaps are defined: front submaps from a front-process and back submaps from a back-process. The back submaps are fed to the front-process to enrich the map elements, facilitating the merging among independent submaps.

#### A. Rumination-inspired Framework

As mentioned above,  $p(x_{0:t}, M_g | F_{0:t})$  is formulated in most of the existing studies. However, in a multiple submap

TABLE II  
THE BIOMIMETIC DEFINITION OF RUMINATION AND RUMI-SLAM

Process	Rumination	RUMI-SLAM
Feed	Food	Captured Image
Swallowing	Rumen	Sliding-window Sampling
Peristalsis	Rumen, Omasum	Additional Submap Building
Regurgitation	Partially Digested Feed	Additional Submaps
Absorption	Abomasum	Submap Global Optimization

VSLAM system, due to the feature of discontinuous observation, we rewrite the probability as Formula 4:

$$p(\mathbf{x}_{0:t}, M_g | \mathcal{F}_{0:t}) = \sum_{i=0}^n p(\mathbf{x}_{0:t}, M_g | \mathbf{z}_i) \cdot p(\mathbf{z}_i) \quad (4)$$

where  $\mathbf{z}_1 = \mathcal{F}_{0:t_0}$ ,  $\mathbf{z}_2 = \mathcal{F}_{t_1:t_2}$ , ...,  $\mathbf{z}_i = \mathcal{F}_{t_k:t_j}$ , and  $n$  is the number of submaps. The sequential process is performed within a part of the observation, that is  $\mathbf{z}_i$ . To satisfy the law of total probability, the observed frame sequence input is assumed to be a perfect incident group. However, due to the untracked images, the exact utilized observation is incomplete, that is  $t_0 \neq t_1$ . Therefore, Formula 4 is not valid and the globally consistent map cannot be achieved. We propose to build additional submaps on untracked images in a back-process, satisfying the assumption of the perfect incident group and the time consumption of challenging frame data association. It is worth noting that the submap building on untracked images is not in realtime, and it is asynchronous to the submap building on  $\mathbf{z}_i$ .

To combine the realtime and non-realtime submap building asynchronously, we propose a rumination-inspired MPMS framework. The framework is a submap-based VSLAM system, namely distributing the submap building and having a centralized submap merging. Inspired by rumination, the biomimetic definition is listed in Tab. II, showing the similarity between rumination and our system. For feeding, the captured images are input to the front-process for realtime submap building; then, a variant density sliding window is designed for swallowing, that divides the captured images between the front-process and the back-process; as for peristalsis, the back-process acts as the role of the rumen and omasum for additional submap building on the challenging images in non-realtime; regurgitation is activated when tracking is lost, that the matched back submap is sent to the front-process; lastly, absorption happens in the centralized submap management, which merges back submaps and front submaps for a globally consistent map.

The proposed MPMS system contains a front-process and a back-process (Fig. 3). The front-process performs realtime visual tracking and local mapping, which is similar to the existing VSLAM systems. In the case of tracking lost, the sampled un-tracked images are sent to the back-process. The back-process is a non-realtime mapping process handling the time-consuming untracked images, and building additional back submaps. In our rumination-inspired framework, the unit of regurgitation is submap. A matched back submap from the back-process enriches the map elements and helps the submap merging. Compared with the existing systems, RUMI-SLAM

enhances across-submap data association using the additional back submap, instead of relying on the loop-closure only.

### B. Asynchronous Submap Building

In RUMI-SLAM, there are two processes for the submap building: the realtime front-process and the non-realtime back-process. For the submap building in the front-process, only a subset of  $M_i$ , denoted as  $sub_i$ , is considered to build the data association  $match_q$  efficiently:

$$sub_i = \{P_k^i \mid |t(p_k^q) - t(P_n^i)| < Th_t \text{ or } |pos(p_k^q) - pos(P_n^i)|_2 < Th_p\} \quad (5)$$

where  $|t(p_q) - t(P_n^i)| < Th_t$  is the condition of the short-term data association,  $t()$  is the timestamp of map elements, and  $Th_t$  is a given threshold.  $|pos(p_q) - pos(P_n^i)|_2 < Th_p$  is the condition of the mid-term data association,  $pos()$  is the position of map elements, and  $Th_p$  is a given threshold. As for the back-process, it uses all the map elements to establish  $match_q$  between the frames, instead of a subset  $sub_i$ . Therefore, the back-process is able to build more data association than the front-process, while it is time-consuming. However, due to the asynchronous design, the back-process is parallel to the front-process, thus the realtime captured image can be localized by the front-process online.

The timing diagram of the asynchronous submap building is shown in Fig. 4, showing the activation mechanism of the proposed MPMS system, involving the front-process, the variant density sampling, the back-process and the optimization. For

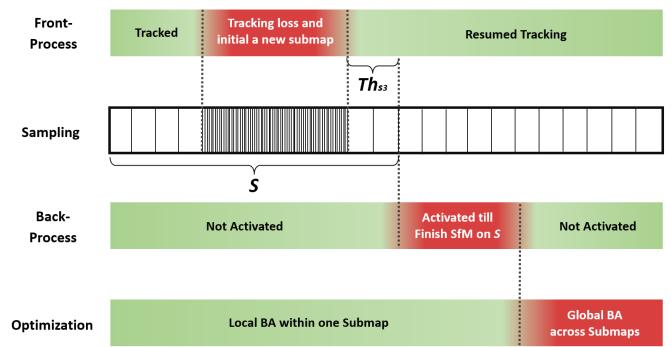


Fig. 4. The top row is the tracking status, that tracking is lost in the “red” stage. The front-process keeps multi-submap building and captures frames. The sampling density is shown in the second row, that sparse sampling is conducted when tracked, and we sample densely when lost. The sampling of  $S$  is stopped by  $Th_{S3}$ . Then, the mapping based on  $S$  in the back-process is activated for additional submap building. After submap regurgitation from the back-process, a global bundle adjustment is performed for submap merging.

robustness, the non-realtime back-process detects  $match_q$  considering all the elements. Since rich map elements have been completed in the back-process, an effective detection can be performed to obtain the data association between front submaps and back submaps. The detail of  $M_{com}$  production is introduced in the next section.

The processing of challenging frames in the back-process is time-consuming. To reduce the time consumption, a sliding window according to the tracking status is designed to reduce

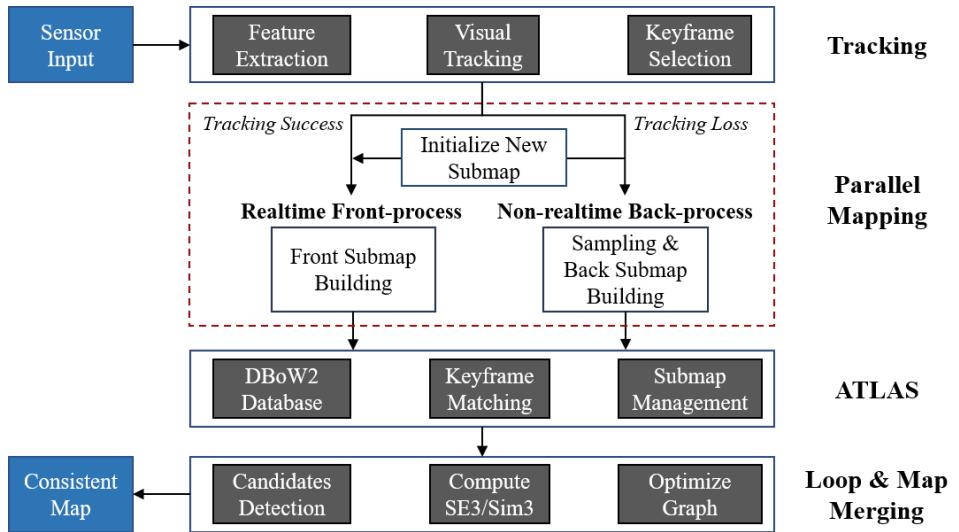


Fig. 3. Framework of the proposed asynchronous MPMS system. Different from the existing work [18], two parallel processes are designed to build the front submap and the back submap (red dotted box). The back submap provides additional elements to improve the submap merging for a globally consistent map.

the frames sent to the back-process. Particularly, we propose a variant density sampling sliding window, which involves three different parts, as shown in the second row of Fig. 4. For the tracked frames, a large interval *inter* and a fixed size *len* are given, to obtain a sample subset  $S_1$ . When tracking in the front-process is lost, an intensive sampling is conducted until tracking is resumed, obtaining a sample subset  $S_2$  of untracked challenging frames. After the tracking is resumed in a new submap, the sampling with *inter* is executed until the number of frames in the front submap is larger than a given threshold  $Th_{S3}$ , obtaining a sample subset  $S_3$ . Finally, the sliding-window sample set is defined as:  $S = S_1 \cup S_2 \cup S_3$ . Ideally,  $S_1$  and  $S_3$  are for the data association between back submaps and front submaps, which reconstruct  $M_{com}$ . The densely sampled frames in  $S_2$  are to build a submap that is missed during tracking loss to maintain the continuity across the front submaps. To further save the computational cost of the back-process, an activation mechanism is proposed. As two parallel processes, the back-process is activated only when untracked images happen. During the back submap building on  $S$ , the front-process keeps realtime tracking and mapping in parallel. After finishing the mapping on  $S$ , the matched back submap is sent to the front-process.

Using the back submaps built on untracked frames (like the image between  $z_1$  and  $z_2$ ,  $z_{12} = F_{t_0:t_1}$ ), we approximate the complete observation of the perfect incident group assumption in Formula 4, as shown in Formula 6:

$$z_1^u \cup z_{12}^u \cup z_2^u \dots \cup z_n^u = F_{0:t}^u \rightarrow F_{0:t} \quad (6)$$

where  $z_*^u$  is the exactly utilized observation, and  $F_{0:t}^u$  is the utilized frames from time 0 to time  $t$ . It is worth noting that the union operation in Formula 6 satisfies the commutative law and is independent of timing, making our non-realtime mapping on supplemental observation feasible. The existing systems are synchronous with a single process, all the observations should be processed in realtime. However,

based on the rewritten formulation, we propose to reconstruct the challenging images  $S_2$  in the back-process and the unchallenging images in the front-process asynchronously. More frames are tracked, and  $F_{0:t}$  can be approximated by  $F_{0:t}^u$ . Therefore, Formula 4 becomes valid for the globally consistent localization and mapping, as shown in Formula 7:

$$p(x_{0:t}, M_g | F_{0:t}^u) \approx p(x_{0:t}, M_g | F_{0:t}) \quad (7)$$

### C. Submap Management

To further improve the submap merging, we design a submap matching method. We build a frame subset  $F^{search}$  that searches  $M_{com}$  intensively and circularly.  $F^{search}$  contains the un-tracked frames in the front-process, and the first and last  $n$  frames of the back submaps. Because  $F^{search}$  are the endpoints of submaps that have a high probability of  $M_{com}$  detection. We match the back submap and the front submap using the timestamp of keyframes. Because the image input of the back-process and the front-process is the same, the matched submap pair should have the keyframe with the same timestamp. After obtaining the candidate submap, we detect  $M_{com}$  by the distance of feature position in the image plane instead of the descriptor distance, since the matched frame is ideally the same as its query one. In addition, due to the low precision of visual reconstruction on the challenging frames, we produce  $M_{com}$  considering the feature matching only. The common geometry-based verification, such as reprojection error, is ignored.

Three steps are designed for submap merging:

**(1) Initial Alignment:** To calculate the initial estimation  $T_{ini}$ , we use Umeyama algorithm [29]. Umeyama algorithm aligns all the matched frame pairs to obtain a Sim(3) solution.

**(2) Optimization:** To perform bundle-adjustment, we optimize  $T_{ini}$  using Formula 8. All the matched elements are considered and the elements in the back submaps set fixed, and a Sim(3) solution is optimized. In Formula 8,  $K$  is the known

intrinsic,  $\mathbf{T}^k$  is the pose of frame  $k$  with respect to its submap, and  $\mathbf{P}_j^i \in \mathcal{M}_{com}$  is the matched map point of the observed feature  $obs$  in another keyframe. As usual,  $\varepsilon(\mathbf{T}) = [s\mathbf{R} \; \mathbf{t}]$ , where  $s$  is the scale,  $\mathbf{R}$  and  $\mathbf{t}$  represent the rotation and the translation, respectively.

$$\mathbf{T}_m = \arg \min_{\mathbf{T} \in Sim(3)} \sum_{i=0} \sum_{j=0} |\mathbf{K} \cdot \varepsilon(\mathbf{T}\mathbf{T}^k) \cdot \mathbf{P}_j^i - obs| \quad (8)$$

**(3) Merging:** After optimization, the optimized  $\mathbf{T}_m$  is applied to all the map elements in the merged submap. Similar to [18], a welding window is used for efficiency. All the matched keyframes construct the welding window, which is merged first and performs a local BA. Finally, we perform post-culling using the method introduced in [5], an operation to cull elements of the back submap and limit the map size.

## V. EXPERIMENTS

### A. Implementation and Experimental Setup

We implement our RUMI-SLAM based on ORB-SLAM3 [18] and COLMAP [30]. ORB-SLAM3 is a submap-based system with the ability of loop-closure detection and submap merging. COLMAP is a SfM system that achieves robust performance at the cost of a long processing time. In terms of the datasets for validations, three monocular image datasets are used: TUM [31], ICL [32] and EuRoC [33]. To demonstrate our advanced robustness, we select the sequences with challenging frames as shown in Fig. 5. As for the computer configuration, we run all the open dataset evaluations and experiments on a desktop computer with an Intel Core i9 (2.80 GHz) CPU, Nvidia 3090 GPU and 64GB memory. The GPU acceleration is only used in the back-process.



Fig. 5. The examples of challenging frames in the selected sequences, including the textureless frames and the blurry frames. Even though these types of frames are few, they make the across-submap data association hard to build.

### B. Experimental Results and Discussion

Three metrics are adopted in the evaluation: (1) the tracking precision calculated by the RMSE, (2) the completeness indicated by tracking rate, calculated by the ratio between the tracked duration and the total duration and (3) the realtime performance indicated by the processing time consumption. In addition, a demonstration of ground navigation is shown with qualitative evaluations.

**1) Back Submap Building :** The back-process is assumed to be more robust than the front-process. To verify this assumption, the evaluation of the back submap building is conducted. We select some sequences with challenging frames to show the robustness of the back-process over the front-process. Most of the selected sequences are rarely included in the existing monocular VSLAM evaluations, due to the low tracking rate and high tracking error. We show the precision, time consumption, the number of frames and the tracking rate in Tab. III. All the results are the mean of 10 executions, the variance is also provided to show the reliability. The sequences with a tracking rate less than 90% are omitted (written as “-”).

For the sequences on which the front-process obtains a tracking rate greater than 90%, the corresponding precision is also higher; for some of the sequences that the front-process cannot complete, the back-process achieves a greater tracking rate. Therefore, the back-process can build additional submaps that the front-process fails; conversely, in the situations where the front-process succeeds, there is no need to launch the back-process to save computational resources. However, in the ICL dataset, both the back-process and the front-process fail on some sequences due to the textureless frames. In the aspect of variance, in most cases, a smaller variance of the back-process indicates a reliable performance. The back-process obtains a big leap in tracking rate and reliability due to the exhaustive search for data association among the frames, which is time-consuming and has a high probability of producing data association.

In terms of time consumption of the back-process, it takes much more time than the front-process in all the cases, which indicates the necessity of decreasing the amount of back-process input by the proposed sampling method. Such a result also indicates that the back-process cannot meet the realtime requirement, and it should be activated only when the front-process fails. As for the front-process, the time consumption of per frame is 0.027s in average, which is similar to the state-of-the-art systems.

**2) Submap Merging and Data Association:** To show the effectiveness of submap merging, an example of additional submap insertion is demonstrated, which is the result on fr1\_room. Given  $inter = 5$  and  $Th_{S3} = 8$ , the sampling with variant density sliding window is shown visually in Fig. 6, as well as the timestamp of tracking lost and tracking resumed. Furthermore, the data association ( $\mathcal{M}_{com}$ ) in RUMI-SLAM is shown in Fig. 7, which is the number of matched features for submap merging. Also, the result without additional submaps and the result with geometry-based verification are plotted for comparison.

The result on sequence fr1\_room from time 937 is illustrated in Fig. 6, in which tracking is lost and then resumed. As is shown, the samples during tracking are sparse, and intensive sampling is conducted when tracking loss occurs. Then, the sparse sampling is resumed after establishing a new submap.  $S_1$  and  $S_3$  of samples in Fig. 6 have overlap with the frames in the front-process, which can produce data association for submap merging.  $S_2$  are the images that cannot be tracked in the front-process, which are reconstructed by the back-process. Also, as shown in Fig. 7, data association is enhanced when the

TABLE III  
A COMPARISON BETWEEN THE BACK-PROCESS AND THE FRONT-PROCESS

Sequence	Back-process			Front-process		
	RMSE (m)	Time (s)	Rate (%)	RMSE (m)	Time (s)	Rate (%)
V1_02	1.262±0.279	4018.15±150.18	99.5 ± 1.7	<b>0.096 ± 0.076</b>	94.20±0.13	97.5±2.8
V1_03	0.930±0.090	2786.61±217.15	100.0 ± 0.0	<b>0.385 ± 0.415</b>	116.48±0.19	94.3±6.8
V2_01	1.724±0.109	4199.70±114.20	100.0 ± 0.0	<b>0.060 ± 0.001</b>	123.85±0.07	95.0±0.1
V2_02	0.953±0.279	2639.23±187.05	98.1 ± 3.9	<b>0.140 ± 0.244</b>	127.91±0.19	97.8±1.1
fr1_floor	0.385±0.197	915.41±58.21	94.4 ± 8.3	-	61.19±0.19	51.4±1.5
fr1_room	0.259±0.081	991.69±87.45	100.0 ± 0.0	-	76.21±1.86	81.7±20.4
fr1_teddy	0.038±0.002	1234.74±67.20	100.0 ± 0.0	-	59.80±0.38	77.6±28.9
fr2_kidnap	<b>0.081 ± 0.004</b>	1113.33±60.34	100.0 ± 0.0	0.531±0.409	75.86±0.82	99.8±0.1
fr2_desk	<b>0.015 ± 0.000</b>	3295.90±214.42	100.0 ± 0.0	0.098±0.267	119.18±0.14	99.1±0.4
fr2_p360	0.127±0.092	975.31±147.80	98.5 ± 4.8	-	100.92±3.35	89.5±12.5
fr2_slam	0.396±0.484	2760.39±198.52	100.0 ± 0.0	-	199.74±2.95	9.5±5.3
fr2_slam2	0.464±0.088	2099.21±51.47	100.0 ± 0.0	-	153.20±4.19	11.2±2.2
fr2_slam3	0.611±0.300	3909.22±433.79	99.9 ± 0.5	-	160.85±4.95	27.9±9.3
fr3_teddy	<b>0.009 ± 0.001</b>	1799.58±74.79	100.0 ± 0.0	0.217±0.258	98.66±0.34	99.7±0.4
fr3_loop	<b>0.013 ± 0.000</b>	1213.89±21.75	100.0 ± 0.0	0.508±0.529	70.19±0.25	99.7±0.2
room_t0	0.243±0.068	902.43±289.46	93.2 ± 0.6	-	91.37±0.19	38.3±0.4
room_t1	-	1365.64±223.48	71.5±10.3	-	51.89±0.50	44.9±15.4
room_t2	0.174±0.006	498.24±35.52	<b>100.0 ± 0.0</b>	-	44.00±0.19	79.7±0.1
room_t3	-	495.05±64.76	75.6±0.1	-	70.36±0.23	48.3±0.8
traj0_p	0.182±0.078	1098.65±58.15	<b>98.5 ± 1.3</b>	-	87.49±0.27	37.7±0.6
traj1_p	-	276.70±45.72	43.1±0.4	-	49.14±0.30	45.1±9.9
traj2_p	0.186±0.000	568.04±30.57	<b>100.0 ± 0.0</b>	-	42.90±0.10	79.7±0.1
traj3_p	0.277±0.196	524.89±33.67	<b>100.0 ± 0.0</b>	-	68.99±0.23	47.0±2.4

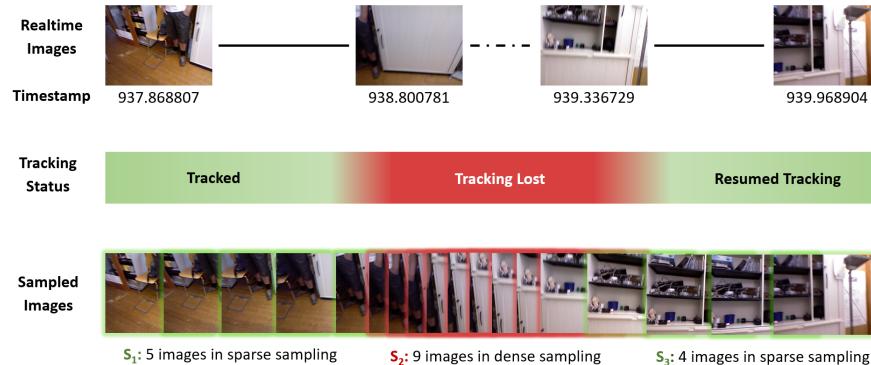


Fig. 6. The result of variant density sampling on fr1\_room from time 937 to 939. The top row is the captured images. The tracking status is indicated by color in the mid-row. The bottom row shows the sampled images. In the example, the number of captured images is 64, and the number of samples is 18.

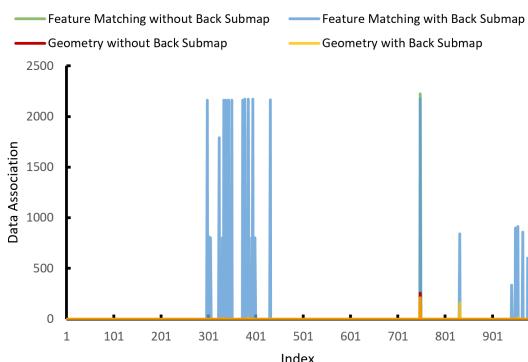


Fig. 7. The number of matched features in the “loop-closing thread”, and the moment of additional submap insertion. Four classes are shown: the result with and without the back submaps, and the matching with feature matching or geometry-based verification [34].

back submap is incorporated. However, the result without back submaps cannot produce data association due to the lack of reconstruction on the un-tracked frames. Also, the result with the geometry-based verification produces less data association, since the front submap on challenging frames is inaccurate and not consistent with the back submap.

Fig. 8 illustrates the number of matched features between two images across a sequence. As shown, besides the matched features between two consecutive frames (the diagonal line), features can also be matched between two distant frames (the upper right part and the lower left part). The feature matching among distant frames indicates the potential long-term data association. RUMI-SLAM achieves global consistency across submaps by building the long-term data association that is lacking in the existing systems.

3) *Comparison:* As an ablation study, we show the results with and without the post-culling (PO) in Tab. IV, respectively. All the results are the mean of 10 executions.

TABLE IV  
A COMPARISON OF THE PROPOSED RUMI-SLAM, ORB-SLAM3, LDSO AND ORB-SLAM2

Sequence	RUMI-SLAM with PO			RUMI-SLAM			ORB-SLAM3			LDSO			ORB-SLAM2		
	Rate	RMSE	Frame	Rate	RMSE	Frame	Rate	RMSE	Frame	Rate	RMSE	Frame	Rate	RMSE	Frame
V1_02	97.2	0.096	262	97.2	0.096	262	97.5	0.096	262	<b>98.3</b>	1.527	741	92.3	<b>0.063</b>	<b>178</b>
V1_03	95.2	0.101	298	95.9	0.098	299	94.3	0.385	299	<b>97.4</b>	0.367	1103	90.1	<b>0.082</b>	<b>226</b>
V2_02	97.1	0.150	<b>281</b>	<b>98.2</b>	0.145	282	97.8	0.140	282	97.8	0.107	889	96.8	<b>0.109</b>	285
V2_03	<b>91.6</b>	<b>0.166</b>	338	91.6	0.166	338	90.6	0.181	336	69.4	-	1220	90.8	0.186	<b>310</b>
fr1_room	90.3	0.150	<b>216</b>	<b>91.8</b>	<b>0.128</b>	702	81.7	-	99	47.2	-	466	33.8	-	81
fr1_teddy	90.7	<b>0.117</b>	<b>223</b>	<b>92.9</b>	0.120	584	77.6	-	292	51.9	-	248	86.5	-	132
fr2_desk	<b>99.2</b>	0.009	<b>272</b>	99.2	0.009	275	99.2	0.098	276	97.5	1.246	398	94.6	<b>0.008</b>	273
fr2_p360	97.3	0.507	<b>314</b>	<b>99.1</b>	<b>0.483</b>	1364	89.5	-	151	98.2	1.694	335	17.3	-	16
fr2_slam	90.6	0.423	<b>119</b>	<b>91.5</b>	<b>0.411</b>	1086	9.5	-	77	92.8	1.643	1244	17.5	-	10
fr2_slam2	<b>93.0</b>	<b>0.477</b>	<b>527</b>	93.4	0.480	601	11.2	-	41	36.6	-	155	20.7	-	16
fr2_slam3	93.2	0.692	<b>334</b>	<b>94.8</b>	<b>0.657</b>	2566	28.0	-	131	72.5	-	619	17.7	-	13
fr3_loop	<b>99.8</b>	0.050	133	99.8	0.050	133	99.7	0.508	133	98.7	0.078	541	99.7	<b>0.016</b>	<b>69</b>
fr3_teddy	99.0	0.229	346	<b>99.8</b>	<b>0.215</b>	347	99.7	0.217	347	89.7	-	614	99.7	0.733	<b>297</b>
room_t0	90.3	<b>0.169</b>	<b>110</b>	<b>90.7</b>	0.246	1436	38.3	-	54	9.4	-	143	33.4	-	49
room_t2	92.1	<b>0.117</b>	<b>147</b>	<b>92.3</b>	0.179	874	45.00	-	30	12.4	-	151	75.5	-	60
traj0_p	96.4	0.614	<b>268</b>	<b>97.0</b>	<b>0.592</b>	1792	37.7	-	52	8.1	-	197	35.6	-	50
traj2_p	94.3	0.236	<b>59</b>	<b>94.6</b>	<b>0.188</b>	214	79.7	-	78	12.4	-	136	73.9	-	54
traj3_p	<b>97.4</b>	0.492	<b>133</b>	97.4	<b>0.426</b>	1224	47.1	-	65	7.2	-	170	54.8	-	51

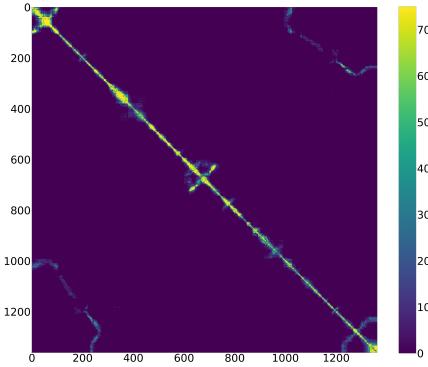


Fig. 8. Feature matching between two images across a sequence. Besides feature matching between two consecutive frames, there is potential data association that can be built for submap merging.

Another key indicator is the tracking rate, which measures the tracking completion of methods. The monocular variant of ORB-SLAM3, ORB-SLAM2 [34] and LDSO are used for comparison, which are the state-of-the-art of indirect and direct methods, respectively. Because RMSE (m) is a statistical result, it is meaningless when the tracking rate is low. We show the RMSE with a tracking rate larger than 90% only. The number of saved frames is also indicated, showing the removal of redundant information.

In most cases, LDSO has the lowest precision, even though it has a higher tracking rate than others in some sequences. Such a performance of LDSO could be caused by the lack of photometric calibration and the robustness of optical flow-based tracking. Also, RUMI-SLAM has comparable performance to ORB-SLAM3 and ORB-SLAM2 on the non-challenging sequences, such as EuRoC dataset, fr2\_kidnap, fr2\_desk, fr3\_loop and fr3\_teddy. However, the strength of RUMI-SLAM is prominent in the sequences that others have a low tracking rate, thanks to the robustness introduced by the back-process. Also, the difference in precision and tracking

rate between the one with and without post-culling is small, while the size of the saved frame with post-culling is decreased by 72.47% on average, showing the efficiency of removing redundant information. It is worth noting that all the tests are run at 30Hz input due to the reduced time consumption by our proposed image sampling method.

To further explain the comparison results, RUMI-SLAM has a front-process that runs submap-based visual tracking and local mapping all the time, and the back-process is inactive during the tracking of unchallenging frames. For the sequences that are completed by the front-process, the non-activated back-process in RUMI-SLAM results in a similar performance to ORB-SLAM3 and ORB-SLAM2. However, for the sequences with challenging frames, especially the ones without loop-closure, the results suggest that RUMI-SLAM achieves a more complete mapping than other existing systems due to the asynchronous processing in our MPMS design. The additional submap from the non-realtime back-process provides more information for long-term data association and submap merging without the requirement of loop-closure, resulting in a higher tracking rate. In addition, the post-culling is shown to reduce storage space while having similar tracking errors and tracking rates. However, in the sequences that others complete, the results with and without post-culling are almost the same, because the back-process is not activated and no duplicated frame in back submaps is removed.

4) *Real-robot experiment:* Lastly, we demonstrate the application of RUMI-SLAM in ground visual navigation. The configuration of the physical experiment platform is shown in Fig. 9. Specifically, we use a Clearpath Jackal mobile robot with an onboard X86 computer for image capturing and motion control. The image acquisition unit is an Intel RealSense collecting 30Hz RGB images using the monocular camera only, with the intrinsics obtained through calibration.

We conduct the physical experiment on our campus. The image sequence contains 22350 images, and the duration is 745.3s. Both the trajectory from RUMI-SLAM (red) and ORB-

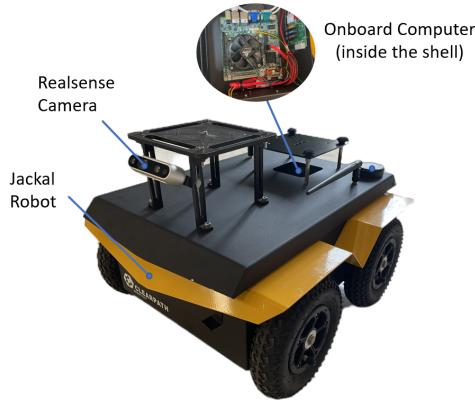


Fig. 9. The mobile robot platform in our physical experiment. A Clearpath Jackal mobile robot with an onboard X86 computer is used. The image acquisition unit is an Intel RealSense.

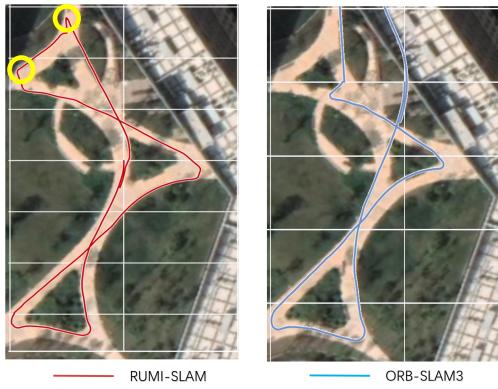


Fig. 10. The estimated trajectory of RUMI-SLAM (red) and ORB-SLAM3 (blue). Tracking is lost in the yellow circle positions. The red trajectory is close to the road, which is more reasonable than the blue one.

SLAM3 (blue) are shown in Fig. 10. RUMI-SLAM is able to achieve a trajectory similar to the road, which is the result in an outdoor environment with challenging conditions such as illumination change, sharp rotation and moving objects. The examples of un-tracked frame in ORB-SLAM3 is shown in Fig. 11, which cannot be tracked by the existing systems due to distant features with little disparity and the moving object. During the experiment, there are two occurrences of tracking loss (yellow circles), which are then recovered by back submaps and submap merging. We achieve 98.77% tracking rate, while it is 83.51% of ORB-SLAM3. In addition, LDSO cannot obtain a valuable result on this dataset due to frequent tracking loss. This physical experiment demonstrates the robustness of RUMI-SLAM over the existing methods and the feasibility of deploying RUMI-SLAM in a laptop equipped with a mobile robot.

## VI. CONCLUSION

In this paper, we propose RUMI-SLAM, which is rumination-inspired and processes images in an asynchronous style. RUMI-SLAM achieves global consistency across submaps without loop-closure. To enhance the long-term data association for submap merging, we build additional submaps



Fig. 11. The challenging frames, including the textureless road surface, the distant features with little disparity and the moving object.

on challenging frames by the back-process, which is time-consuming but robust. To reduce the time consumption of the back-process, a variant density sliding window is introduced to limit the amount of input. Due to the asynchronous MPMS framework, the realtime requirement is met by the front-process, and the robustness to challenging frames is achieved by the back-process. The experimental results demonstrate a high tracking rate and globally consistent map on the challenging sequences while having comparable accuracy to the state-of-the-art VSLAM systems. Also, our advanced tracking robustness has been verified in challenging situations on the physical experiment, which cannot be completed by the existing VSLAM systems.

Our most important contribution is the rumination-inspired framework, which can be extended to the implementation of edge-cloud collaboration in the future.

## REFERENCES

- [1] W. Chen, L. Zhu, X. Lin, L. He, Y. Guan, and H. Zhang, "Dynamic strategy of keyframe selection with pd controller for vslam systems," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 115–125, 2022.
- [2] R. Li, S. Wang, and D. Gu, "Deepslam: A robust monocular slam system with unsupervised deep learning," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3577–3587, 2021.
- [3] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed certifiably correct pose-graph optimization," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2137–2156, 2021.
- [4] W. Chen, L. Zhu, Y. Guan, C. R. Kube, and H. Zhang, "Submap-based pose-graph visual slam: A robust visual exploration and localization system," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6851–6856.
- [5] M. J. M. M. Mur-Artal, Raúl and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [6] Z. Dai, X. Huang, W. Chen, L. He, and H. Zhang, "A comparison of cnn-based and hand-crafted keypoint descriptors," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2399–2404.
- [7] L. Yu, J. Qin, S. Wang, Y. Wang, and S. Wang, "A tightly coupled feature-based visual-inertial odometry with stereo cameras," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 4, pp. 3944–3954, 2023.
- [8] L. Zhang, W. Yan, and H. Li, "An efficient online loop closure detection system with local spatial co-occurrence information," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 8, pp. 8174–8183, 2023.
- [9] (2022) Rumi-slam and the evaluation tools. [Online]. Available: <https://github.com/Changfei-Fu>
- [10] 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.
- [11] Y. Chen, S. Huang, R. Fitch, and J. Yu, "Efficient active slam based on submap joining, graph topology and convex optimization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5159–5166.

- [12] J. Yu, J. Tong, Y. Xu, Z. Xu, H. Dong, T. Yang, and Y. Wang, "Smmr-explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8779–8785.
- [13] R. Giubilato, M. Vayugundla, M. J. Schuster, W. Strzl, A. Wedler, R. Triebel, and S. Debei, "Relocalization with submaps: Multi-session mapping for planetary rovers equipped with stereo cameras," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 580–587, 2020.
- [14] P. Pinis and J. D. Tards, "Large-scale slam building conditionally independent local maps: Application to monocular vision," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1094–1106, 2008.
- [15] R. Elvira, J. D. Tards, and J. Montiel, "Orbslam-atlas: a robust and accurate multi-map system," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6253–6259.
- [16] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.
- [17] S. Macenski and I. Jambrecic, "Slam toolbox: Slam for the dynamic world," *Journal of Open Source Software*, vol. 6, no. 61, p. 2783, 2021.
- [18] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tards, "Orb-slam3: An accurate open-source library for visual, visualeinertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [19] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.
- [20] X. Gao, R. Wang, N. Demmel, and D. Cremers, "Ldso: Direct sparse odometry with loop closure," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2198–2204.
- [21] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena, "maplab 2.0 c a modular and multi-modal mapping framework," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 520–527, 2023.
- [22] M. Labb   and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.
- [23] N. Keivan and G. Sibley, "Asynchronous adaptive conditioning for visual-inertial slam," *The International Journal of Robotics Research*, vol. 34, no. 13, pp. 1573–1589, 2015.
- [24] H. Huang, W.-Y. Lin, S. Liu, D. Zhang, and S.-K. Yeung, "Dual-slam: A framework for robust single camera navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4942–4949.
- [25] P. J. Weimer and M. B. Hall, "The potential for biomimetic application of rumination to bioreactor design," *Biomass and Bioenergy*, vol. 143, p. 105822, 2020.
- [26] C. Birmingham and T. Firoz, "Rumination in eating disorders: literature review," *Eating and Weight Disorders-Studies on Anorexia, Bulimia and Obesity*, vol. 11, no. 3, pp. e85–e89, 2006.
- [27] "A decision-tree model to detect post-calving diseases based on rumination, activity, milk yield, bw and voluntary visits to the milking robot," *Animal*, vol. 10, no. 9, pp. 1493–1500, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1751731116000744>
- [28] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.
- [29] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.
- [30] J. L. Sch  nberger and J.-M. Frahm, "Structure-from-Motion Revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [31] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.
- [32] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *2014 IEEE international conference on Robotics and automation (ICRA)*. IEEE, 2014, pp. 1524–1531.
- [33] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.
- [34] R. Mur-Artal and J. D. Tard  s, "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.

**Weinan Chen** born in 1991, he is currently an associate professor in the Guangdong University of Technology. During his Ph.D. degree, he spent a year at University of Alberta, Edmonton, Canada, as a Visiting Scholar. After a year of post-doctoral training at The Chinese University of Hong Kong, he joined the Southern University of Science and Technology as the "Principal Outstanding Postdoctoral Fellow". His research interests include SLAM and mobile robots.



**Changfei Fu** received the B.E. degree in electronics and information engineering from China University of Mining Technology. He is currently a postgraduate in Southern University of Science and Technology. His current research interests include slam and machine vision.



**Lei Zhu** born in 1992, is currently a Ph.D. student at School of Electro-mechanical Engineering, Guangdong University of Technology, China. Her research interests include sensor calibration and robot localization.



**Shing Yan Loo** received the MSc degree from the Universiti Putra Malaysia (UPM). He is currently enrolled on a Dual Ph.D. program offered by UPM and University of Alberta. His research interest lies at the intersection of visual SLAM and deep learning.



**Hong Zhang** born in 1959, is currently a professor at Department of Electronic and Electrical Engineering, Southern University of Science and Technology in Shenzhen, China. His research interests include robotics, computer vision, and image processing. Within the international robotics community, his activities include a variety of roles in ICRA and IROS communities. He is a Fellow of IEEE and a Fellow of the Canadian Academy of Engineering.

