Exploration with Global Consistency Using Real-Time Re-integration and Active Loop Closure

Yichen Zhang, Boyu Zhou, Luqi Wang and Shaojie Shen

Abstract— Despite recent progress of robotic exploration, most methods assume that drift-free localization is available, which is problematic in reality and causes severe distortion of the reconstructed map. In this work, we present a systematic exploration mapping and planning framework that deals with drifted localization, allowing efficient and globally consistent reconstruction. A real-time re-integration-based mapping approach along with a frame pruning mechanism is proposed, which rectifies map distortion effectively when drifted localization is corrected upon detecting loop-closure. Besides, an exploration planning method considering historical viewpoints is presented to enable active loop closing, which promotes a higher opportunity to correct localization errors and further improves the mapping quality. We evaluate both the mapping and planning methods as well as the entire system comprehensively in simulation and real-world experiments, showing their effectiveness in practice. The implementation of the proposed method will be made open-source for the benefit of the robotics community.

I. INTRODUCTION

Autonomous exploration, which sends robots to completely map an unknown environment, has been a popular topic in robotics research, because it is a fundamental problem for a variety of applications such as aerial photography, geographic mapping, disaster management, and search and rescue. During the past decades, many algorithms have been proposed to improve exploration performance [1]–[3].

Despite recent progresses, most of the current methods concern only about the exploration efficiency and give less consideration about the map quality. In particular, many of them assume that perfect localization is available throughout the exploration, so accurate reconstruction is easily obtainable as well. Unfortunately, in practice, accumulating drift of onboard state estimation is inevitable in a long run. The drift often has a negative impact on the reconstructed map, such as causing significant distortions. Consequently, the resulting map may be of poor quality, which is unsatisfactory for many applications that require accurate reconstruction. Therefore, to enable a broader range of applications, the mapping algorithm should be able to rectify the distortion caused by localization drift explicitly. Moreover, previous methods seldom account for potential opportunities to detect loops, although the map quality is highly dependent on the detection of loop closure. Instead, detecting loop closure during exploration is mostly passive and depends on whether the
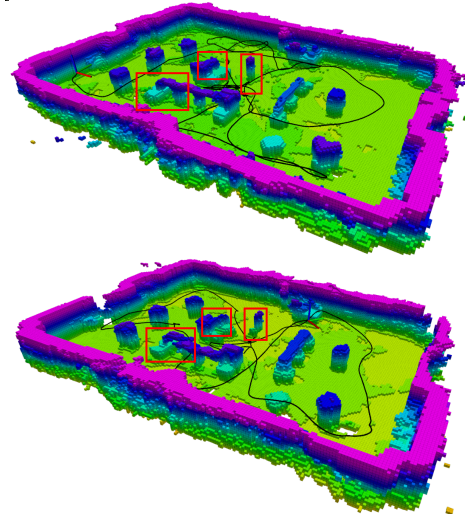
Fig. 1. The quadrotor autonomous exploration tests are conducted in a complex indoor scene with size $20 \times 14 \times 3\text{m}^3$. From top to bottom: Actual scene of experiment field; Our exploration results; FUEL [1] results. Significant differences are marked with red box.

robot happens to pass a previously visited place. Hence, the exploration algorithm should seek for more opportunities on loop closing without sacrificing much efficiency to improve map quality further.

In order to solve the two issues mentioned above, we propose a systematic solution for autonomous exploration, which enables the generation of accurate and globally consistent maps. First, we introduce a TSDF-based mapping framework with real-time re-integration and a frame pruning mechanism based on a set cover formulated selection criterion. When loop closure is detected and drift is eliminated by the localization module, the map distortion caused by localization drift is rectified by re-integrating the keyframes with updated camera poses, resulting in a globally consistent map. To enable real-time re-integration on the onboard computer, we novelly apply a keyframe selection method based on set cover formulation to prune the redundant frames. Then, we utilize space partitioning to incrementally solve the set cover problem, increasing the real-time performance. Moreover, we adopt an active loop closing strategy in exploration planning

to allow a higher opportunity to detect loops, which further reduces localization drift and enhances map quality.

Extensive experiments are conducted in both benchmarked simulations and the real world to validate the performance of our proposed method. In the simulation, we compare our proposed framework with the results from other mapping approaches. The results show that our proposed method outperforms other works, especially under severe odometry drift. In real world tests, our proposed method generates a more globally consistent exploration result, comparing with the state-of-the-art exploration system [1]. The contributions of this work can be summarized in the following:

1) A globally consistent mapping framework with online re-integration and frame pruning mechanism, which can rectify the map distortion.
2) An active loop closing planning strategy which allows more loops to be detected thus further improves the mapping quality.
3) Benchmark comparison experiments and field tests are conducted to evaluate the performance of our proposed method. The source code of our work will be published.

II. RELATED WORK

A. Autonomous Exploration

Robotic exploration has been investigated over the years. Some works aim at quick coverage [4, 5], while others emphasize precise mapping [6, 7]. Frontier-based approaches are presented earliest in [8] and assessed more comprehensively later in [9]. In [2] a stochastic differential equation-based strategy is used for 3D environments. Unlike [8] choosing the closest frontier as the next target, [4] chooses the frontier inside the FOV that minimizes the velocity change. The strategy is advantageous for a high flight speed and shows higher efficiency than [8]. [10] presented a differentiable formulation of information gain, which is useful for gradient-based optimization of the exploration path.

Another major category is sampling-based approaches, which produce viewpoints randomly to explore the space. These approaches are closely related to next best view (NBV) [11], where views are computed repeatedly to model a scene completely. [12] grows RRTs and executes the most informative edge repeatedly. Uncertainty [13], visual importance [14], inspection [15] and history [16, 17] are considered later under this framework [12].

Approaches combining the advantages of frontier-based and sampling-based approaches are presented in [1, 7, 18]–[22]. [18, 19] generate global paths based on frontiers and sample paths locally. A two-level framework [21] computes exploration paths coarsely at the global scale and finely around the robot. In [22], sparse topological graphs are built to provide high-level guidance. [1] proposed a hierarchical planning framework based on the incrementally extracted frontiers, which shows high-performance exploration in complex environments.

A common issue of existing methods is that they seldom consider the rectification of map under odometry drift, except
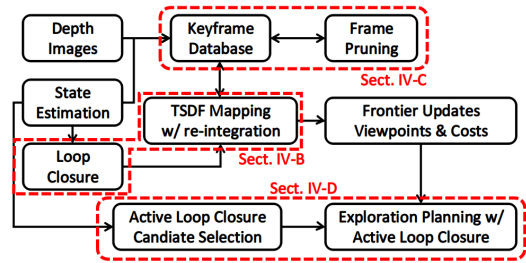

Fig. 2. System overview of our proposed exploration framework.

[23], which may not produce globally consistent map in practice. In this work, we deal with this problem systematically.

B. Globally Consistent Dense Mapping

Researchers have suggested a number of approaches to rectify distorted map when loop-closures are detected. One idea is to attach a deformable mesh [24] to a pose graph representing the past trajectory of the camera. Alternatively, a deformable collection of surfels may be maintained [25]. Although these approaches provide a richer understanding of the occupied space, the free space and unknown space are not distinguished, so they are not suitable for robotic exploration where information of unknown space is essential.

Another approach is building a collection of submaps to present the global map. The relative transformation of each submap is corrected when localization is rectified by loop-closure, which can maintain map consistency efficiency. Several works have shown the efficacy of such approaches, including [26]–[29] One limitation of these methods is that they put the burden on the motion planner, which has to query map information from multiple submaps. Besides, they can not eliminate distortion completely under severe localization drift.

[30] proposes a re-integration method to maintain a globally consistent map, which stores sensor measurements and associated sensor poses to rectify the map upon localization correction. It produces higher reconstruction quality, but requires significant computational resources. [31] improves the computational efficiency by a large margin. However, it does not explicitly represent unknown space. In contrast to submaps, these works provide constant query time for motion planning module. In this work, a mapping method based on re-integration is adopted and further optimized by a frame pruning technique, enabling real-time mapping in exploration.

III. SYSTEM OVERVIEW

An overview of our proposed systematic solution for autonomous exploration can be found in Fig. 2. With the depth measurement and onboard state estimation, the mapping module can carry out a globally consistent reconstruction for exploration planning. The re-integration (Sect. IV-B) of rectified point cloud plays a key role in eliminating map distortion caused by pose drift during the flight. We use a frame pruning method (Sect. IV-C) formulated as a set cover problem to enable online re-integration. Besides considering frontiers in space, possible loop closure viewpoints (Sect.

IV-D) are also included in autonomous exploration planning for better global consistency.

IV. METHODOLOGY

A. TSDF-based 3D Mapping

The mapping module of the proposed method is based on Truncated Signed Distance Field (TSDF), whose construction method is similar to [30, 32, 33]. The key difference is that our framework allows real-time and globally consistent reconstruction by adopting the re-integration (Sect.IV-B) and frame pruning (Sect.IV-C) methods.

We take the point cloud generated from the depth image as an input frame. For each frame, ray-casting is performed from camera position ^wC to each point ^wP in the point cloud and all voxels along the ray are updated with new distance value sdf and weight w. The distance value sdf can be computed as the length difference of the vector $\overrightarrow{^wC^wP}$ and $\overrightarrow{^wC^wV}$, where ^wV is the point that the center of voxel v projected on the ray $\overrightarrow{^wC^wP}$. Note that the sdf value is bounded to range $[-\delta, \delta]$ as [30] to support de-integration process, where $\delta = t\sigma_{map}$ is the truncated distance and t is a scaling factor and σ_{map} is the map resolution.

$$sdf = \min\{\delta, \max\{-\delta, \| \overrightarrow{^wC^wP} \| - \| \overrightarrow{^wC^wV} \|\}\} \quad (1)$$

For the TSDF weighting method, a constant weighting policy can be applied to the TSDF construction for simplicity. While aiming to improve the quality of details in the mapping result, a weighting strategy considering both depth value z and behind-surface voxels, which is proposed by [32, 34], is applied.

$$w = \begin{cases} \frac{1}{z^2}, & -\sigma_{map} < sdf \leq \delta \\ \frac{1}{z^2}\frac{\delta+sdf}{\delta-\sigma_{map}}, & -\delta \leq sdf < -\sigma_{map} \end{cases} \quad (2)$$

Now for a TSDF voxel v along the ray, its distance value $v.sdf$ and weight $v.w$ can be updated as

$$v.sdf = \frac{v.sdf * v.w + sdf * w}{v.w + w}, \ v.w = v.w + w \quad (3)$$

Thanks to that TSDF construction inherently supports the reversible updating, we can de-integrate redundant or drifted frames from the mapping result to maintain global consistency by re-integration (See IV-B). To de-integrate a frame from the map, v can be updated in reverse as

$$v.sdf = \frac{v.sdf * v.w - sdf * w}{v.w - w}, \ v.w = v.w - w \quad (4)$$

One important function of the mapping framework is to provide information about known and unknown space to our exploration planner. Traditionally, the voxel is regarded as known if its TSDF weight is not zero, i.e. has been observed by at least once. To efficiently execute frontier-based exploration planning and maintain a good observation quality for mapping at the same time, we set a weight threshold $\tau_w > 0.0$ as the criterion for known and unknown voxels recognition, where

$$v.state = \begin{cases} \text{Unknown}, & 0.0 < v.w \leq \tau_w \\ \text{Known}, & \tau_w \leq v.w \end{cases} \quad (5)$$

By applying this strategy, we can make sure every place in the map can get enough observations by a proper τ_w and larger τ_w means more observations are needed to change the area to a known area in exploration planning.

B. Online Re-integration

When loop closure is detected, a 4-DOF pose graph optimization is performed with multiview constraints from loop closing frames and camera poses on the pose graph are relocalized [35], in which the global consistency of state estimation is obtained.

In order to rectify the map distortion correspondingly, all keyframes with correction on its corresponding camera pose are de-integrated from the map with its drifted pose and re-integrated with its rectified pose [30]. Keyframes are selected from input frames in the keyframe database based on the method introduced in IV-C. The number of keyframes that are going to be re-integrated depends on the size of the subgraph rectified by loop closure. A large rectified subgraph of the rectified pose graph will generate a long list of keyframes to be re-integrated, which will cause a load burst on computing and might block the normal integration process. To reduce its influence, keyframes that need re-integration are put into a waiting list and then distributed to following regular integration cycles. Besides, for safety flight concerns, local maps have a higher demand for map consistency. Therefore, keyframes in current and neighbor blocks have utmost priority during the re-integration process. The blocks are obtained by space partitioning illustrated in IV-C.

C. Redundant Frame Pruning

The objective of the frame pruning process is to remove redundant point cloud frames that provide much duplicate information about map details but cumber the re-integration process, as shown in Fig. 3. To identify the representative frames that do not contain superfluous information and enable real-time re-integration, we form this selection criterion as the well-studied un-weighted Set Cover Problem (SCP).

The universe set U is a voxel grid in the explored area with resolution σ_{scp}. As this voxel grid only served for the checking of space observed by each frame, the resolution requirements here is lower than mapping which needs clear details about obstacles. We set $\sigma_{scp} = 2\sigma_{map}$ to accelerate visibility check for each frame without significant changes on frame's coverage. For frame f, the voxels $\{v_{f,1}, v_{f,2}, ..., v_{f,n_f}\}$ in U within its Field of View (FOV) are gathered as a subset $S_f \subseteq U$ with cost $c_f = 1.0$ for an un-weighted set cover problem. The cost effectiveness α for frame f is defined as follows, where $f.n_{new}$ is the number of new voxels the frame f covered.

$$\alpha_f = \frac{c_f}{f.n_{new}} \quad (6)$$

In the classic greedy algorithm for set cover problem, the goal will be formed as that for k frames, we need to find a keyframe set $K \subseteq \{S_1, S_2, ..., S_k\}$ with minimum total

Algorithm 1 Keyframe Selection

Input: Point cloud frames list $\mathcal{L}$
Output: Keyframe list $\mathcal{L}_K$

1: $gain \leftarrow \infty$
2: **while** $gain > \tau_{gain}$ **do**
3:     **for each** $f \in \mathcal{L}_{\mathcal{R}}$ **do**
4:         **if** f is not selected **then**
5:             $f.n_{all} \leftarrow$ number of grids f covers
6:             $f.n_{covered} \leftarrow 0$
7:             **for each** $g \in f.coveredGridsList$ **do**
8:                 **if** $g.visbilityCount == 0$ **then**
9:                     $f.n_{covered} \leftarrow f.n_{covered} + 1$
10:                 **end if**
11:             **end for**
12:             $f.n_{new} \leftarrow f.n_{all} - f.n_{covered}$
13:         **end if**
14:     **end for**
15:
16:     $f^* \leftarrow \arg\min_{f} \frac{f.cost}{f.n_{new}}$
17:     $gain \leftarrow f^*.n_{new}$
18:     $f^*.selected \leftarrow true$
19:     $\mathcal{L}_K.push(f^*)$
20:
21:     **for each** $g \in f^*.coveredGridsList$ **do**
22:         **if** $g.visbilityCount > 0$ **then**
23:             $g.visbilityCount \leftarrow g.visbilityCount - 1$
24:         **end if**
25:     **end for**
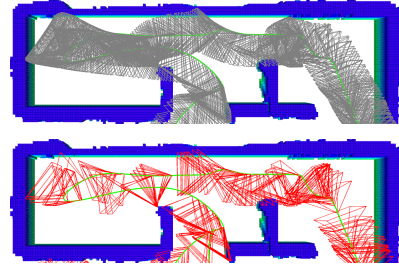26: **end while**

Fig. 3. Illustration of the frame pruning. The pyramids represent the frame's FOV. In the top image, inessential frames (grey) will decelerate the re-integration process. In the bottom picture, keyframes (red) are selected from above frames by adopting set cover formulation.
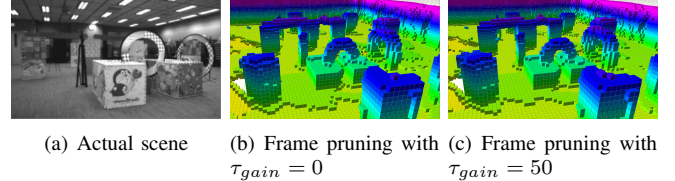

(a) Actual scene   (b) Frame pruning with $\tau_{gain} = 0$ (c) Frame pruning with $\tau_{gain} = 50$

Fig. 4. The frame pruning and a small value of τ_{gain} has little effect on the map quality. The number of frames in (b), (c) is 544 and 456 respectively. The total number of frames is 1468.

cost $c_{total} = \sum_{i \in [1,k]} c_i$, such that $\cup_{i \in [1,k]} S_i = U$. We make several changes based on the classic algorithm to make the implementation compatible and practical with our framework. First, we make a minimum number of observations n_{ob} for each voxel and each voxel in U will be initialized with a visibility counter with value n_{ob}. Only when the visibility counter of one voxel is deducted to zero, this voxel will be marked as covered and more observation on this voxel will be disposed. This is because covering each voxel in U just once is not enough, which will be easily disturbed by noise or drifted pose. With this manner, the n_{new} of frame f can be calculated as the difference between total number of its visible voxels and how many voxels among these are already marked as covered. Second, we modify the termination criterion from $\cup_{i \in [1,k]} S_i = U$ to the gain of one cycle , which is counted as the number of new voxels covered by this frame in U, is less than a certain threshold τ_{gain}. Ideally, τ_{gain} is set to 0 to ensure a full coverage. However, in practice, the solution got from the classic greedy algorithm usually have a number of frames providing little new information compared with tens of thousands voxels covered by each frame, i.e. $f.n_{new} \ll f.n_{all}$. Such a few new voxels have almost no effect on the mapping result as shown in Fig. 4. Therefore, we assign τ_{gain} with a small value ($\sim \frac{10}{\sigma_{scp}}$), which can reduce the keyframe amount and

accelerate re-integration process without loss of quality. With the initial visibility count for each voxel set to 1 and τ_{gain} set to 0.0, our keyframe selection will degenerate to the classic greedy method. More details can be found in Algorithm 1.

Solving the set cover problem for the frames on the entire path is extremely time-consuming after a long flight. We adopt space partitioning and divide the map into equal-size axis-aligned blocks and execute set cover solver only on blocks with new frames. For N frames, this can reduce the time complexity from $O(\log N)$ to $O(M_{new} \log \frac{N}{M_{all}})$, where M_{all} and M_{new} are number of all blocks and blocks with new frames and $M_{new} \ll M_{all}$ typically.

D. Exploration Planning with Active Loop Closure

Since the mapping result relies on globally consistent localization, which requires more loop closure to be detected, we propose an exploration planning with Active Loop-Closure (ALC) technique based on our previous approach [1]. The key idea is to find an exploration tour that efficiently visits all frontier clusters frontier clusters [8] as well as potential loop-closing sites.

To identify the candidate location where the quadrotor may have an opportunity to detect a loop-closure, consecutive historical viewpoints along the flight path are grouped into clusters, as illustrated in Fig. 5. We empirically constrain the size of each cluster in order to distinguish distinct regions previously visited by the quadrotor. To efficiently find the nearby viewpoints, which are considered by our exploration planning, we also maintain a KD-Tree for the historical viewpoints[1] in the flight path (Line 1, Algorithm 2).

Firstly, we consider the active loop closure clusters into the global tour planning [1]. In [1], a global tour passing only frontier clusters is generated, while in our method

---

[1]Though the viewpoint belongs to (x, y, z, ϕ), the construction and query of KD-Tree here only consider the viewpoint's position.

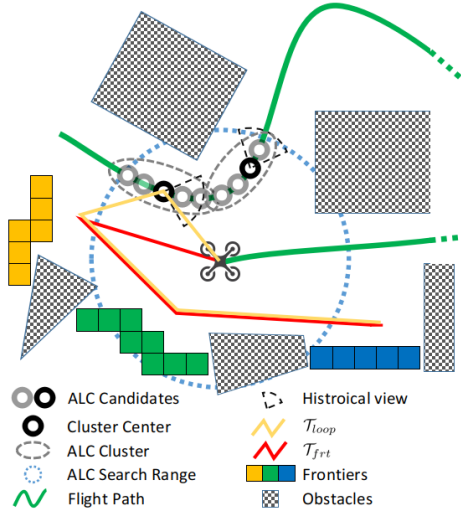|		
◎◯ ALC Candidates	⌐⌐ Histroical view	
◯ Cluster Center	$\mathcal{N}$ $\mathcal{T}_{loop}$	
⌐⌐ ALC Cluster	$\mathcal{N}$ $\mathcal{T}_{frt}$	
⌐⌐ ALC Search Range	▮▮ Frontiers	
$\mathcal{N}$ Flight Path	▨▨ Obstacles	

Fig. 5. An illustration of the proposed active loop closure method. The UAV will search loop closure candidate within a certain range and evaluate the cost of different global tours.

we also consider previously visited viewpoints. If a loop closure cluster candidate exists nearby the quadrotor, we choose a cluster with minimum cost from current state to its center viewpoint. Next, an optimal tour $\mathcal{T}_{loop}$ that visits the candidate first and all frontier clusters subsequently is found by solving an Asymmetric Traveling Salesman Problem (ATSP). To this end, a cost matrix M_{loop} containing the cost information among all frontiers and the selected ALC cluster is computed. The cost calculation of two viewpoints utilizes the time lower bound required to move between them proposed in [1]. In order to avoid large detour which will significantly reduce the exploration efficiency, the optimal tour $\mathcal{T}_{frt}$ considering only frontier cluster is also generated and compared with $\mathcal{T}_{loop}$. To find $\mathcal{T}_{frt}$, a cost matrix M_{frt}, which contains the cost information among all frontiers and current state is also generated. Note that although an extra tour $\mathcal{T}_{loop}$ is computed comparing with [1], it does not introduce significant overhead, as M_{loop} can be obtained trivially by slightly modifying M_{frt}, as shown in Line 7-10, Algorithm 2. Then the next goal of flight is decided by comparison between the total cost of original global tour with only frontiers C_{frt} and the cost of global tour containing the selected ALC cluster C_{loop}.

$$C_{loop} \leq (1+\epsilon)C_{frt} \qquad (7)$$

Since active loop closure will usually take a detour which reduce the exploration efficiency, different ϵ denotes a trade-off between the exploration efficiency and global consistency. The greater ϵ means the planner will tolerate more cost spending on loop closure and maintain a more globally consistent map.

As the global planning only decide if the active loop closure is needed and select the center viewpoint, more viewpoints in the ALC clusters besides the center viewpoint are taken into consideration for local viewpoints refinement based on the graph search-based approach in [1], which further improves the path quality and allow faster exploration. Finally, a minimum-time trajectory is generated towards the next viewpoint along the refined path through an efficient

---

Algorithm 2 Active Loop Closure Planning

Input: Flight path $\mathcal{P}$, current position p, cost matrix M_{frt} and global tour $\mathcal{T}_{frt}$ with only frontiers, frontier list $FrtList$

Output: Global exploration tour $\mathcal{T}_G$

1: $loopClusters \leftarrow nearestNeighbors(p)$
2: **for each** $c \in loopClusters$ **do**
3:    $c.cost \leftarrow computeCost(p, c)$
4: **end for**
5: $c^* \leftarrow \arg\min_c c.cost$
6:
7: $M_{loop} \leftarrow M_{frt}$
8: **for each** $frt \in FrtList$ **do**
9:    $M_{loop}(0, frt.index) \leftarrow computeCost(c^*, frt)$
10: **end for**
11: $\mathcal{T}_{loop} \leftarrow solveAsymmetricTSP(M_{loop})$
12:
13: $C_{frt} \leftarrow 0, C_{loop} \leftarrow 0$
14: $computeTotalCost(p, \mathcal{T}_{frt}, M_{frt}, C_{frt})$
15: $computeTotalCost(p, \mathcal{T}_{loop}, M_{loop}, C_{loop})$
16: $C_{loop} \leftarrow C_{loop} + c^*.cost$
17: **if** $C_{loop} < (1+\epsilon)C_{frt}$ **then**
18:    $\mathcal{T}_G \leftarrow \mathcal{T}_{frt}$
19: **else**
20:    $\mathcal{T}_G \leftarrow \mathcal{T}_{loop}$
21: **end if**
22:
23: **function** $computeTotalCost(p, \mathcal{T}, M, C)$:
24:    $vpLast \leftarrow p$
25:    **for each** $vp \in \mathcal{T}$ **do**
26:       $C \leftarrow C + M(vpLast, vp)$
27:       $vpLast \leftarrow vp$
28:    **end for**

---

gradient-based optimization framework [36], enabling safe and agile navigation. The quadrotor will be detached from local ALC clusters if no loop closure is successfully detected within a certain period, which prevents distraction from exploration the task. Note that we may evaluate the possibility of loop-closure for historical viewpoints as [37] does, which is left as a future work.

V. EXPERIMENT RESULT & COMPARISON

A. Benchmarked Simulation

Extensive experiments are conducted in simulation environment. We compare our mapping result obtained from autonomous exploration with other representative works. For a better evaluation on the accuracy and robustness of the methods, we set four different level of odometry drift from s1 to s4 by applying a Gaussian Noise $v_{xyz} \sim N(\mu_{xyz}, \sigma^2_{xyz})$ and $\omega_\phi \sim N(\mu_\phi, \sigma^2_\phi)$ on each odometry updating cycle with frequency 200Hz. Details of the drift parameter setting can be found in Table I. The dynamic parameters are set as $v_{max} = 2.0$ m/s, $a_{max} = 2.0$ m/s^2 and $r_{max} = 0.9$ rad/s.

Fig. 6. Mapping result of exploration in simulation environment ($30 \times 16 \times 3$ m^3) under severe odometry drift (s4). From left to right: the ground truth point cloud, Voxblox, C-blox, our method w/o frame pruning and our method. It can be found that our method can maintain a good global consistency during the exploration flight. If the loop closure result is close to the actual localization, our reconstruction result is approximate to the ground truth.

TABLE I
DRIFT PARAMETER SETTING

param (m/s)	s1	s2	s3	s4
μ_{xyz}	0.0	0.0	0.0	0.0
σ^2_{xyz}	0.0	2e-2	5e-2	8e-2
μ_ϕ	0.0	0.0	1e-3	1.5e-3
σ^2_ϕ	0.0	2e-2	5e-2	8e-2

TABLE II
ACCURACY COMPARISON

Method	RMSE (m)			
	s1	s2	s3	s4
Voxblox [32]	0.092	0.084	0.169	0.266
C-blox [26]	0.109	**0.081**	0.147	0.157
Ours(w/o frame pruning)	**0.089**	0.091	0.098	**0.085**
Ours	**0.089**	0.087	**0.094**	0.087

We uniformly sample a collection of zero-level iso-surface points on the TSDF. For each of the point, a nearest point from the ground truth point cloud is found to compute the distance error.

As the result shown in Fig. 6 and Table II, four methods have a similar performance under negligible odometry drift. When it comes to odometry drift, the result of Voxblox [32] distorts the most as it does not include rectifying global map according to input loop closure results. C-blox [26] can support such map rectification by correct each submap with rectified camera localization. However, it still has a higher RMSE mainly because it assumes state estimation is consistent within a submap, which is not reasonable under large drift. The result of our method w/o frame pruning, i.e. re-integrate all frames, shows a similar result as ours, but its real-time performance is poor on onboard computer.

We also compare the time cost for exploration w/ and w/o active loop closure under s3 drift in Table I, which is 115s(w/o ALC) and 124s(w/ALC). The exploration flight is shown in Fig. 7. With $\epsilon = 0.2$ in Sect. IV-D, the time of exploration planning with ALC is only 8% more than that w/o ALC but it gains 3 more opportunity to detect loop closure. The final mapping result shows that exploration w/ ALC has superior performance under odometry drift.

B. Field Test

In order to further evaluate the performance of our proposed approach, comparative field experiments are conducted in a large classroom with cluttered environment between the FUEL [1] and ours. The bounded size of the exploration area is $20 \times 14 \times 3$ m^3. With the installation of a RealSense depth camera D435i and a DJI Manifold 2-C onboard computer powered by Intel Core i7-8550U, the quadrotor platform fully
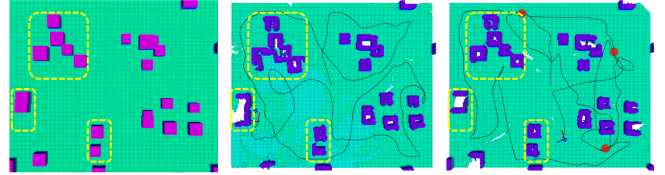

Fig. 7. Left: Ground truth ($20 \times 16 \times 3$ m^3), Middle: w/o ALC; Right: w/ ALC. Successful loop closure is marked as red dot.
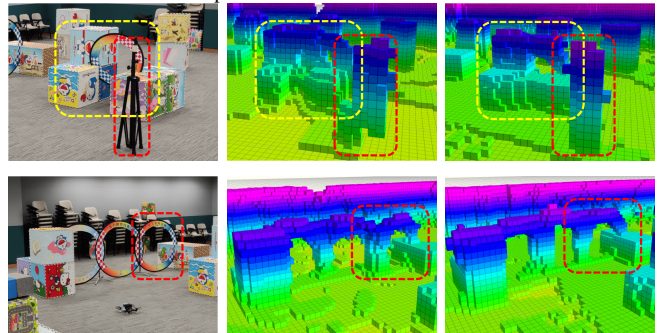

Fig. 8. From left to right: the intercepted actual scene from Fig. 1, the reconstruction result from FUEL [1] and ours. For the first row, the result from [1] has strong distortion on the ring and the tripod. On the second row, our method reconstructs three coplanar rings well.

relies on onboard sensors and computation to perform the exploration task without any external devices. The onboard state estimation and loop closure method is VINS-Fusion [35]. The linear velocity, acceleration and yaw rate limits of both methods are set to $v_{max} = 1.0$ m/s, $a_{max} = 0.5$ m/s^2 and $r_{max} = 0.6$ rad/s.

A global map and local details comparison between FUEL and ours is shown in Fig. 1 and 8. Though loop-closure of localization is enabled for both methods, the mapping result from FUEL still has several distortions. In our result, a globally consistent map is maintained. The flight path in Fig. 1 labeled with black line shows the effect from the active loop closure planning. This experiment demonstrates that our novel approach has the ability to maintain a global consistent mapping result during autonomous exploration.

VI. CONCLUSIONS

In this paper, we propose a systematic mapping and planning framework for exploration under odometry drift, which can maintain the global consistency of the exploration map. The mapping is based on an online re-integration method with frame-pruning to enable real-time performance. Moreover, we consider the possible loop closure viewpoints actively into the exploration planning to improve the mapping result. The proposed method is evaluated by extensive experiments in both benchmarked simulation and real-world environment. The results show the effectiveness of our proposed method.

References

[1] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, 2021.

[2] S. Shen, N. Michael, and V. Kumar, "Stochastic differential equation-based exploration algorithm for autonomous indoor 3d exploration with a micro-aerial vehicle," *Intl. J. Robot. Research (IJRR)*, vol. 31, no. 12, pp. 1431–1444, 2012.

[3] M. Keidar and G. A. Kaminka, "Efficient frontier detection for robot exploration," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 215–236, 2014.

[4] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.(IROS)*. IEEE, 2017, pp. 2135–2142.

[5] M. Dharmadhikari, T. Dang, L. Solanka, J. Loje, H. Nguyen, N. Khedekar, and K. Alexis, "Motion primitives-based path planning for fast and agile exploration using aerial robots," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*. IEEE, 2020, pp. 179–185.

[6] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.

[7] S. Song and S. Jo, "Online inspection path planning for autonomous 3d modeling using a micro-aerial vehicle." in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, 2017, pp. 6217–6224.

[8] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[9] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Auton. Robots*, vol. 33, no. 4, pp. 427–444, 2012.

[10] D. Deng, R. Duan, J. Liu, K. Sheng, and K. Shimada, "Robotic exploration of unknown 2d environment using a frontier-based automatic-differentiable information gain measure," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2020, pp. 1497–1503.

[11] C. Connolly, "The determination of next best views," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, vol. 2. IEEE, 1985, pp. 432–435.

[12] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon" next-best-view" planner for 3d exploration," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*. IEEE, 2016, pp. 1462–1468.

[13] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty-aware receding horizon exploration and mapping using aerial robots," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4568–4575.

[14] T. Dang, C. Papachristos, and K. Alexis, "Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2526–2533.

[15] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3d exploration and surface inspection," *Auton. Robots*, vol. 42, no. 2, pp. 291–306, 2018.

[16] C. Witting, M. Fehr, R. Bähnemann, H. Oleynikova, and R. Siegwart, "History-aware autonomous exploration in confined environments using mavs," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[17] C. Wang, D. Zhu, T. Li, M. Q.-H. Meng, and C. W. de Silva, "Efficient autonomous robotic exploration with semantic road map in indoor environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2989–2996, 2019.

[18] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, "Information-theoretic planning with trajectory optimization for dense 3d mapping." in *Proc. of Robot.: Sci. and Syst. (RSS)*, vol. 11, 2015.

[19] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, "Efficient autonomous exploration planning of large-scale 3-d environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1699–1706, 2019.

[20] Z. Meng, H. Qin, Z. Chen, X. Chen, H. Sun, F. Lin, and M. H. Ang Jr, "A two-stage optimized next-view planning framework for 3-d unknown environment exploration, and structural reconstruction," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1680–1687, 2017.

[21] C. Cao, H. Zhu, H. Choset, and J. Zhang, "Exploring large and complex environments fast and efficiently."

[22] F. Yang, D.-H. Lee, J. Keller, and S. Scherer, "Graph-based topological exploration planning in large-scale 3d environments," *arXiv preprint arXiv:2103.16829*, 2021.

[23] L. Schmid, V. Reijgwart, L. Ott, J. Nieto, R. Siegwart, and C. Cadena, "A unified approach for autonomous volumetric exploration of large scale environments under severe odometry drift," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4504–4511, July 2021.

[24] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," 2012.

[25] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph." Robotics: Science and Systems, 2015.

[26] A. Millane, Z. Taylor, H. Oleynikova, J. Nieto, R. Siegwart, and C. Cadena, "C-blox: A scalable and consistent tsdf-based dense mapping approach," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.(IROS)*. IEEE, 2018, pp. 995–1002.

[27] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto, "Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 227–234, 2019.

[28] R. Wagner, U. Frese, and B. Bäuml, "Graph slam with signed distance function maps on a humanoid robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2691–2698.

[29] N. Fioraio, J. Taylor, A. Fitzgibbon, L. Di Stefano, and S. Izadi, "Large-scale and drift-free surface reconstruction using online sub-volume registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4475–4483.

[30] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundle-fusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[31] L. Han and L. Fang, "Flashfusion: Real-time globally consistent dense 3d reconstruction using cpu computing." in *Robotics: Science and Systems*, vol. 1, no. 6, 2018, p. 7.

[32] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[33] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*. IEEE, 2011, pp. 127–136.

[34] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, "Real-time camera tracking and 3d reconstruction using signed distance functions," in *Robotics: Science and systems (RSS) conference 2013*. Robotics: Science and Systems, 2013.

[35] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot. (TRO)*, vol. 34, no. 4, pp. 1004–1020, 2018.

[36] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.

[37] E. M. Lee, J. Choi, H. Lim, and H. Myung, "Real: Rapid exploration with active loop-closing toward large-scale 3d mapping using uavs," *arXiv preprint arXiv:2108.02590*, 2021.