Decentralized LiDAR-inertial Swarm Odometry

Fangcheng Zhu*, Yunfan Ren*, Fanze Kong, Huajie Wu, Siqi Liang, Nan Chen, Wei Xu, Fu Zhang

Abstract—Accurate self and relative state estimation are the critical preconditions for completing swarm tasks, e.g., collaborative autonomous exploration, target tracking, search and rescue. This paper proposes a fully decentralized state estimation method for aerial swarm systems, in which each drone performs precise ego-state estimation, exchanges ego-state and mutual observation information by wireless communication, and estimates relative state with respect to (w.r.t.) the rest of UAVs, all in real-time and only based on LiDAR-inertial measurements. A novel 3D LiDAR-based drone detection, identification and tracking method is proposed to obtain observations of teammate drones. The mutual observation measurements are then tightly-coupled with IMU and LiDAR measurements to perform real-time and accurate estimation of ego-state and relative state jointly. Extensive real-world experiments show the broad adaptability to complicated scenarios, including GPS-denied scenes, degenerate scenes for camera (dark night) or LiDAR (facing a single wall). Compared with ground-truth provided by motion capture system, the result shows the centimeter-level localization accuracy which outperforms other state-of-the-art LiDAR-inertial odometry for single UAV system.

I. INTRODUCTION

Multi-robot system, especially aerial swarm system, has great potential in many aspects, such as autonomous exploration [1], [2], target tracking [3], search and rescue, etc. Thanks to its great team cooperation capability, a swarm system can complete missions in complex scenarios even in degenerate environment for a single drone. For a single drone system, accurate self-localization [4]–[7] is the precondition for obstacle avoidance [8], [9] and flight control [10]. For an aerial swarm, robust, accurate self and relative state estimation [11] play a similarly vital role. For a swarm to fulfill cooperative tasks, each drone in the swarm needs to perform real-time, precise self-localization with low drift, and needs to be aware of other drones’ states at all times.

Most recent works on aerial swarm mainly focus on collaborative planning [12]–[14], while research on self and relative state estimation of swarm system is still a large gap. For outdoor scenes, GPS and RTK-GPS are usually adopted [15], [16]. While for GPS-denied places such as indoor scenarios, motion capture system [14], visual (-inertial) odometry [13], [17] or LiDAR (-inertial) odometry [1], [18] methods are much preferred. Besides, Ultra-WideBand (UWB) is also adopted in [12], [19] to produce robust localization result. Although RTK-GPS, motion capture and UWB with anchors [20] have great accuracy, they are dependent on cumbersome, extra devices and would lead to laborious installation. Moreover, these mentioned methods render the whole system centralized, which is fragile to single-point-of-failure (SPOF). Camera is widely used due to its light weight, low cost and rich color information, but it is vulnerable to inadequate illumination and is lack of direct depth measurement, leading to high computational complexity when computing 3D measurements. Apart from that, the range of depth measurements of camera is quite limited. Anchor-free UWB is also low-cost and light-weight. However, it can only offer one dimensional distance measurement and the accuracy is quite limited (usually meter-level), which may decrease the overall accuracy of the whole swarm system.

Compared with camera and anchor-free UWB, LiDAR can provide accurate 3D measurements and is also robust to illumination changes. Even in dark scenes, e.g. underground tunnels, LiDAR can help to perform accurate localization and mapping [17]. Recent years, some cost-effective solid-state LiDARs such as Livox [1] emerged in the market, which significantly expanded the scope of LiDAR-based researches. In this paper, we propose a robust, real-time and decentralized swarm odometry based on LiDAR-inertial measurements. The main contributions are highlighted as follows:

1) A novel 3D LiDAR-based drone detection, identification and tracking method, providing accurate 3D mutual observation measurements used for self and relative state estimation. Each drone is attached with reflective tapes, such that a teammate drone can be reliably and efficiently detected from the reflectivity values of LiDAR points. The detected teammate drone is then tracked by a Kalman filter and matched with trajectories received from the shared network to obtain the teammate’s identity and its initial relative state.

2) A fully decentralized, robust, data-effective and computationally-efficient ego-state and relative state estimation framework. In this framework, each drone only has to exchange its ego-state and teammate observation information, which requires extremely low communci-
tion bandwidth. The mutual observation measurement is tightly-coupled with LiDAR and IMU measurements under Error State Iterated Kalman Filter (ESIKF) framework to achieve robust, accurate, and simultaneous self and relative state estimation.

3) Low-cost, decentralized hardware and software framework. The sensors and computing resource are totally on-board, including Livox LiDAR, IMU and on-board computer. The communication is achieved with Ad-Hoc network available on standard network module.

4) Implementation and validation of the proposed method in extensive real-world experiments, in both indoor (see Fig. [1], outdoor and degenerate scenes.

II. RELATED WORKS

State estimation for aerial swarm systems is much more complicated than that of single drone system because each drone not only needs to perform accurate self-localization with low drift, but also needs to estimate teammates’ states in its own global frame. Among these swarm systems, cameras and LiDARs are two mainstream sensors.

Many existing works on swarm state estimation are based on image measurements. In [13], [22] each drone in the swarm runs an independent VIO to achieve ego-state estimation. The ego-state is then exchanged so that each drone can estimate the relative state of other teammates given the relative transformation between their initial pose (global extrinsic). The global extrinsic is usually calibrated offline or estimated online with good initial guess [11]. Since the VIO of each drone is completely independent, it may lead to large relative state estimation error as the system runs, due to imprecise global extrinsic or VIO drift caused by long-time running. To constrain the relative state during the online running of swarm systems, mutual observation based methods are vigorously developing. Nguyen et al. [23] propose visual-inertial-multi-drone localization system, MAVNet is used to detect other teammate drones. To further enhance the mutual observation, Xu et al. [11] propose a visual-inertial-UWB relative state estimation system utilizing YOLOv3-tiny for teammate drone detection, and UWB module is adopted to provide distance constraints as a complementary sensor of camera. These learning-based drone detection methods usually need long time of prior training, resulting in increase of the total time-consumption of system implementation. One of the biggest restrictions of vision-based methods is the range and quality of depth measurement are quite limited, which narrows down their applications in practical cases, e.g. in large-scale, outdoor environments.

By contrast, LiDAR can provide accurate and long-range depth measurements, bringing many new opportunities for swarm state estimation. In [18], [20], [24], 3D LiDAR-based place recognition (loop closure) constraints are utilized to improve state estimation accuracy. However, since cloud register of raw or segmented points are inevitable in place recognition, the computation load is quite heavy. Thus, all [18], [20], [24] need a centralized master agent to receive data transmitted by other agents and finish those computationally intensive tasks, i.e., cloud register or map merging, increasing the risk of single-point-of-failure. To ease the burden of communication and computation, Huang et al. [25] propose a decentralized multi-robot system using Scan Context descriptor, permitting data-efficient exchange of LiDAR descriptors among robots. Apart from map and descriptor exchange mentioned above, mutual observation is another important unique feature of swarm system, which is a light-weight type of information, avoiding large communication and computation load. Wasik et al. [26] propose a laser-based multi-robot system, laser range finders are used for each robot to estimate the distances and angles to other robots. However, the used 2D LiDARs are not applicable to drones considered in this paper which fly in 3D spaces.

Compared with learning-based detection methods [11], [23], we attach reflective tapes on each drone and leverage the reflectivity measured by LiDAR sensors to detect teammate drones. Different from [11], [13], we propose a course-to-fine calibration method to acquire accurate global extrinsic transformations without any prior initial guess. The rough calibration result acquired by trajectory matching is fed into ESIKF for further online refinement along with ego-state estimation. Compared with centralized systems in which the drones exchange map information [13], [20], [24], our system is fully decentralized which would suffer no SPOF issue, and communication-efficient which exchanges ego-state and mutual observation information only.

III. METHODOLOGY

A. Problem Formulation

Aiming to assist understanding of the proposed system, we define some important notations here. \( x_i, \hat{x}_i, \bar{x}_i, \tilde{x}_i \) represent the ground-true, predictive, updated, and measured state of drone \( i \) respectively. \( t_{ik} \) denotes the timestamp of the \( k \)-th ESIKF state update of drone \( i \).

Considering an aerial swarm system consisting of \( N \) drones and each one carries a LiDAR and an inertial measurement unit (IMU), the state estimation is decomposed into two parts. The first part is ego-state estimation. Each drone, here we choose drone \( i \) for demonstration convenience, needs to estimate its own position \( G_i p_i \in \mathbb{R}^3 \) and attitude \( G_i R_i \in SO(3) \) in its global frame \( G_i \). Since the IMU measurements are coupled with unknown and time-varying bias, the gyroscope bias \( b_{v_i} \in \mathbb{R}^3 \) and accelerometer bias \( b_{a_i} \in \mathbb{R}^3 \) should be calibrated. Moreover, gravity vector \( g \) in each global frame \( G_i g \in \mathbb{R}^3 \) should also be estimated online. The self position \( G_i p_i \), attitude \( G_i R_i \), velocity \( G_i v_i \), bias \( b_{v_i, a_i} \), and gravity vector \( G_i g \) constitute the ego-state, whose estimation is exchanged among the swarm system. Then, to estimate the relative state of other teammate drones based on the exchanged ego-state information, each drone needs to estimate the global extrinsic transformation \( G_i T_{G_j} = (G_i R_{G_j}, G_i p_{G_j}) \in SE(3) \) with respect to the teammate drone \( j \) (\( j = i, \cdots, N, j \neq i \)). Finally, to provide mutual observations of other teammates, each drone \( i \) needs to detect, identify, and track any other teammate \( j \), whose observed position in the self body frame is denoted by \( b_j p_i \).
B. Framework Overview

Each drone in the swarm system runs an independent copy of the same framework. The overview of the framework copy run by the $i$-th drone is shown Fig. 2. The first stage is initialization (Sec. III-C), including detection and temporary tracking for potential teammate drones (Sec. III-C.1). Once the motion of temporarily tracked object is sufficiently excited, the trajectory matching (Sec. III-C.2) starts. If the trajectory of the temporarily tracked object is successfully matched with a teammate’s trajectory transmitted by wireless communication, the object would be considered as the observation of the corresponding teammate and be labeled with the teammate identity (ID), the temporary tracker hence becomes a teammate tracker (Sec. III-D.1). The observed position of teammate identity (ID), the temporary tracker hence becomes associated with the closest predicted position. If no valid association is found, i.e., error of predicted position and cluster position is too large, the tracker re-clusters the raw points into the predicted position and cluster position. If no valid object is clustered, the tracker will propagate to the next step. The tracker will be killed if it has propagated for too many steps without an update. Note that since the points in predicted region are far less than all input points, the time consumption of re-clustering would be significantly decreased.

2) Teammate Identification using Trajectory Matching: By rejecting invalid clusters and tracking real potential teammates, the trajectory of each temporary tracker is accumulated for subsequent identification. Since all drones in the proposed swarm system would exchange their estimated ego-state (in their own global frame) with others, the teammate identification and global extrinsic can be acquired by trajectory matching as follows:

$$\arg\min_{G_i, T_{G_j}} \sum_{k=1}^{K} ||G_i P_{m,k} - G_i T_{G_j} G_i P_{b_j,k}||$$

(1)

where $G_i P_{m,k} \in G_i \mathcal{T}_m$ represents the $k$-th position in the tracked position trajectory $G_i \mathcal{T}_m$ of the $m$-th object and $G_i P_{b_j,k} \in G_i \mathcal{T}_j$ represents position received from drone $j$. Considering possible short-term communication disconnection, some data of $G_i P_{b_j}$ might be lost. Thus, we only pick $G_i P_{m,k}$ that has close timestamp with $G_j P_{b_j,k}$ to participate in trajectory matching. Besides, to avoid large computing
time due to too much data, we use a sliding window of the most recent $K$ positions for matching.

Since no unique transformation can be determined from (1) if the involved trajectories are straight lines [27], the trajectories of those tracked objects are constantly assessed by $\text{TrajExcited}(G_i, T_m)$ in Line 6 until sufficient information is collected. Let $G_i P_m$ represent the centroid of $G_i, T_m$. $\text{TrajExcited}(G_i, T_m)$ assess the excitation (shape) of $G_i, T_m$ by computing the singular values of matrix $H \in \mathbb{R}^{3 \times 3}$:

$$H \triangleq \sum_{\kappa=1}^{K} (G_i P_m, \kappa - G_i P_m) \cdot (G_i P_m, \kappa - G_i P_m)^T \quad (2)$$

If the second largest singular value is larger than a given threshold, the trajectory is fully excited, which is qualified to do $\text{TrajMatching}(G_i, T_j, G_i, T_m)$ in Line 8. This function solves (1) which has well-studied closed-form solution [27]. The matching is performed with each received teammate drone’s trajectory until the matching error is smaller than a given threshold, indicating that the object $m$ is essentially the observation of teammate $j$, and the solution of (1) gives an initial estimation of the global extrinsic $G_i, T_{G_i}$, which is then refined online using the ESIKF in Sec. III-D.3. After identification, a temporary tracker becomes teammate tracker with the corresponding drone ID, which will be sequentially tracked as Sec. III-D.1. The initialization pipeline is illustrated in Fig. 3.

D. Decentralized LiDAR-inertial State Estimation

The fully decentralized state estimation of the proposed swarm system is a tightly-couple iterated Kalman filter inherited from FAST-LIO2 [4], but further incorporates the mutual observation constraints to improve ego-state estimation accuracy, and includes online refinement of inter-drone global extrinsic transformations.

1) Teammate Tracking: After the detection and identification of a teammate drone, a teammate tracker is obtained and the global extrinsic is initially calibrated by trajectory matching. The teammate tracker is similar to a temporary tracker, but with two key differences. The first difference lies in the prediction. Both temporary and teammate tracker predict state using constant velocity model, but a teammate tracker predicts the state based on the velocity received from the corresponding teammate (and the most recent estimation of the corresponding global extrinsic transformation), instead of from an ego-estimated velocity in a temporary tracker (see Sec. III-C.1). The second difference lies in state update when no teammate observation is available. In a temporary tracker, if there is no valid observation, $e.g.$, the teammate is out of FoV, the tracker will propagate for a few steps and then terminate as explained in Sec. III-C.1. In a teammate tracker, it will use the teammate's estimated velocity received from the network, after transforming to the self-drone’s global frame using the most recent global extrinsic transformation obtained by Sec. III-D.3 to continue the state update.

2) State Prediction: Denote $\tau$ as the IMU measurement index, the discrete state transition model is shown below:

$$x_{i, \tau+1} = x_{i, \tau} + \Delta t \cdot f_i(x_{i, \tau}, u_{i, \tau}, w_{i, \tau}) \quad (3)$$

where $\Delta t$ is the time interval of two consecutive IMU measurements, $x_{i, \tau}$ denotes ground-truth state at IMU measurement’s timestamp $t_{\tau}$. The state vector $x_i$, discrete state transition function $f_i$, noise $w_i$, and input $u_i$ are defined as:

$$x_i \triangleq [G_i R_{b_i} G_i p_i G_i v_i b_i a_i g_i \ldots] \in \mathcal{M}$$

$$f_i \triangleq [\omega_m, \omega_n, n_m, n_n, a_m, a_n, g_m, g_n, \ldots] \in \mathbb{R}^{15 \times 1 \ldots}$$

$$w_i \triangleq [\nu_{m}, \nu_{n}, n_{v_m}, n_{v_n}, \nu_{a_m}, \nu_{a_n}, \nu_{g_m}, \nu_{g_n} \ldots]$$

where $\nu_m, a_m$ represent the IMU measurements of drone $i$, the meaning of each element in state vector $x_i$ is introduced in Sec. III-A, the state manifold $\mathcal{M}$ are defined in (4) and its dimension is $18 + 6 \times (N - 1)$.

$$\mathcal{M} \triangleq \mathbb{SO}(3) \times \mathbb{R}^{15} \times \ldots \times \mathbb{SO}(3) \times \mathbb{R}^{3} \times \ldots \quad (4)$$

**Algorithm 1: Initialization**

**Input:** Raw LiDAR points in body frame $b_iP_{all}$, drone $i$’s odometry $G_i T_b_i$, time interval of LiDAR input $\Delta t$, drone $j$’s position trajectory $G_j T_j$, given threshold of trajectory matching residual $\theta_{thr}$.

**Output:** Global extrinsic transformation $G_i T_{G_j}$, tracked position of drone $j$ in drone $i$’s global frame $G_i p_j$.

1. $G_i P_b = \text{ReflectivityFiltering}(b_i P_{all})$
2. $G_i P_m = \text{EuclideanClustering}(b_i P_{all})$
3. for $m = 1 : M$ do
4. $G_i P_m = \text{TemporaryTracker}(G_i T_b_i, b_i P_{all}, b_i P_m)$
5. $G_i T_m, \text{PushBack}(G_i P_m)$
6. if $\text{TrajExcited}(G_i, T_m)$ then
7. for $\tau = 1 : N$, $j \neq i$ do
8. $\text{res, } T = \text{TrajMatching}(G_i, T_j; G_i, T_m)$
9. if $\text{res} < \theta_{thr}$ then
10. $G_i T_j = T$
11. $G_i p_j = G_i P_m$
12. break;
13. end
14. end
15. end
In [3], we used the notation $\boxplus/\boxminus$ defined in [28] to compactly represent the "plus" on the state manifold. Specifically, for the state manifold $SO(3) \times \mathbb{R}^n$ in (III-D.2), the $\boxplus$ operation and its inverse $\boxminus$ are defined as

$$[R_a \boxplus r_b] = [\text{RExp}(r) : R_1 \boxminus R_2] = [\text{Log}(R_2^T R_1)]$$

where $R, R_1, R_2 \in SO(3), r \in \mathbb{R}^3, a, b \in \mathbb{R}^n$, $\text{Exp}() : \mathbb{R}^3 \mapsto SO(3)$ is the exponential map on $SO(3)$ [28] and $\text{Log}() : SO(3) \mapsto \mathbb{R}^3$ is its inverse logarithmic map.

The state prediction step of the $i$-th drone’s state under ESIKF framework is implemented once receiving a new IMU measurement as follows:

$$\hat{x}_{i,t+1} = \hat{x}_{i,t} \boxplus (\Delta t \cdot f(\hat{x}_{i,t}, u_{i,t}, \tau); \hat{x}_0 = \hat{x}_{i,k-1} \quad (5)$$

3) Error State Iterative State Update: The update step is implemented iteratively at the end of new LiDAR scan $t_{ik}$, fusing point-cloud measurements and mutual observation measurements (if any). Once receiving a new scan, motion compensation would be performed to obtain undistorted points, and point-to-plane distance will be computed to generate point-cloud residuals. Details of the motion compensation can be referred to [4]. Denote each motion undistorted point projected into global frame using propagated ego-state as $G_j \hat{p}_n$, denote $u_n$ as the normal vector of the corresponding plane, on which a point $G_j q_n$, the point residual is denoted as $z_{p,n}$.

Apart from point-cloud residual, one main contribution of this paper is 3D LiDAR-based mutual observation measurements, which are used to construct new constraints to improve state estimation accuracy and to render the swarm system robust to degenerate scenes. For drone $i$, denote $z_{ao,ij}$ as the active observation residual arising from $b_j \hat{p}_b$ (the active observation measurement w.r.t drone $j$), see Sec. III-D.1, and denote $z_{po,ij}$ as the passive observation residual arising from $b_j \hat{p}_b$ (the passive observation measurement w.r.t drone $j$), then the residual block of drone $i$ is composed as:

$$z_i = [\cdots, z_{p,n}, \cdots, z_{ao,ij}, \cdots, z_{po,ij}, \cdots]^T$$

$$z_{p,n} = u_n^T(G_j \hat{p}_n - G_j q_n)$$

$$z_{ao,ij} = G_j T_{b_j}^{-1} G_j \hat{p}_b - b_j \hat{p}_b$$

$$z_{po,ij} = G_j T_{b_j}^{-1} G_j \hat{p}_b - b_j \hat{p}_b \quad (6)$$

The state will be iteratively updated until convergence, the details can be referred to [4]. After convergence, the map will be incrementally updated using ikd-tree [29], an efficient data structure for map management. The relative state estimation is completed by projecting the ego-state information transmitted by teammate drones using the updated global extrinsic transformations.

IV. EXPERIMENTS

The experiment platform contain three drones, and each one carries a Livox LiDAR (drone 1 and drone 3 carry Livox Mid360 and drone 2 carries Livox Avia with a smaller FoV), a pixhawk flight controller (BMI055 6-axis IMU inside), an onboard computer Intel NUC with CPU i7-10710U. Each drone is attached with reflective tapes for easy detection (See Fig. 4). For each drone, the time offset and extrinsic between LiDAR and IMU are calibrated by [30], and the time among different drones are synced beforehand based on Network Time Protocol (NTP). All drones fly in manual mode using model predictive control [10].

A. Indoor Flights

Five flights in indoor scenes are performed to evaluate the localization accuracy. Motion capture system is introduced as the ground-truth. As far as we know, since there is no other 3D LiDAR-based state estimation methods for aerial swarm systems, we conduct ablation study by comparing localization accuracy of our method to FAST-LIO2 [3] (state-of-the-art LiDAR-inertial odometry for single drone system). From the position RMSE shown in Table I, it can be seen that for drone 2, the localization error of the proposed method fusing mutual observation measurements is obviously smaller than that of FAST-LIO2. The reason lies in the limited FoV of Livox Avia LiDAR, which is only $70.4^\circ \times 77.2^\circ$. The small FoV makes the LiDAR easier to drift due to lack of structural features, especially in indoor scenes. In this case, the mutual observation measurements can provide strong constraints to acquire accurate localization results. While for drone 1 and drone 3, thanks to the $360^\circ$ horizontal FoV of Mid360 LiDAR, there are sufficient point-cloud constraints to keep FAST-LIO2 work well. Thus, fusing mutual observations does not improve the localization accuracy significantly. For all experiments the time interval of LiDAR input is 100 ms, the average time-consumption per scan of the proposed method and FAST-LIO2 is 17.5 ms and 14.8 ms. It proved the proposed swarm odometry can be implemented in real-time and has approximate computational efficiency to FAST-LIO2 (a fast odometry for single UAV system).

B. Degenerate Environment Flights

The first degenerate scene is data 05 shown in Table I where drone 2 carrying a Livox Avia LiDAR faces a smooth wall which imposes insufficient constraints to determine the full state. As a result, the single-agent odometry FAST-LIO2 completely degenerates with large RMSE. While for the proposed swarm system, drone 2 is observed by its teammate drone 1 and drone 3 (the passive observation measurement). Fusing passive observation measurements transmitted from drone 1 and drone 3 in [6] will still lead to stable and accurate state estimation. The point-cloud map, mutual observation measurements are illustrated in Fig. 5.

The second degenerate scene is a smooth cuboid corridor. Similar to the smooth wall, observations of a single LiDAR


TABLE I

|        | Drone 1 | Drone 2 | Drone 3 |
|--------|---------|---------|---------|
| Data   | FAST-LIO2 | Proposed Swarm | FAST-LIO2 | Proposed Swarm | FAST-LIO2 | Proposed Swarm |
| 01     | 0.0407  | 0.0390  | 0.0464  | 0.0673  | 0.0359  | 0.0347  |
| 02     | 0.0296  | 0.0280  | 0.0428  | 0.0692  | 0.0305  | 0.0307  |
| 03     | 0.0393  | 0.0372  | 0.0442  | 0.0836  | 0.0320  | 0.0313  |
| 04     | 0.0355  | 0.0341  | 0.0413  | 0.0827  | 0.0298  | 0.0310  |
| 05     | 0.0242  | 0.0255  | 0.0356  | 0.0844  | 0.0327  | 0.0319  |

The position error (RMSE) of FAST-LIO2 and the proposed swarm system (unit: meter).

Fig. 5. Degenerate smooth wall experiment. The white points in (a) and (b) refer to current input points. (a) View of drone 2 with point-cloud map constructed by FAST-LIO2, severe drift occurs and the map is messy. (b) View of drone 2 with point-cloud map constructed by the proposed method fusing passive observation measurements $b_3^{p12}$ and $b_3^{p2}$. Yellow boxes means that teammate drones are out of FoV, teammate trackers are updated fusing ego-state estimation transmitted by teammates. (c) Picture of the three drones. (d)(e) View of drone 1 and drone 3 respectively and their observations of teammate drones (the blue boxes).

Do not provide sufficient constraints for pose determination, so FAST-LIO2 suffers from drifts. While for a swarm system, when the first drone flies into the corridor, other drones still hover at the entrance (where enough structural features exist for their state estimation) and provide passive observation measurements for the first drone. After the first drone flies through the corridor, it hovers at the end, offering passive observation measurements for rest of the drones, so they could also fly through the corridor. Thus, the whole swarm can fly through a smooth corridor without any degeneration. From Fig. 5 it can be seen the proposed method is robust to this type of degenerate scene, the map can recover more structural details than FAST-LIO2. More visual illustrations can be found in our video.

C. Decentralized Exploration

To validate that the proposed swarm system is robust to observation lost and can calibrate the global extrinsic transformations, an large-scale decentralized exploration experiment is conducted. Three drones are randomly placed in front of a mansion and then take off. After the take-off, they perform some random flights during which each drone detects, tracks, and identifies each other all automatically and without any prior extrinsic information. After that the drones fly in different directions to explore different areas. After flights finish, the point-cloud map of each drone can be merged together offline using the estimated global extrinsic. By transforming point-cloud map of drone 2 and drone 3 to drone 1’s global frame, the merged map is shown in Fig. 7. The consistent merged map shows the capability of accurate global extrinsic estimation without any initial value. Another point should be noted is, since the drones fly in different regions, there is no mutual observation for a long time, e.g., for drone 2 the average distance without mutual observation is 88.58 m. But still, when the teammate drone returns to drone 2’s FoV, the error between the detected teammate position and the predicted position by the teammate tracker (based on the received teammate ego-state, see Sec. III-D) is only 0.17 m, which indicates the high accuracy of the estimated global extrinsic transformation. More visual illustrations are shown in our video.

V. CONCLUSION AND FUTURE WORK

This paper proposed a fully decentralized and accurate LiDAR-inertial swarm odometry. A novel 3D LiDAR-based drone detection, identification and tracking method was proposed to perform fully autonomous initialization and robust relative state estimation. Mutual observation measurements are tightly-coupled with IMU and point-cloud measurements under ESIKF framework, fulfilling real-time and accurate ego-state estimation, which can keep robust even in degenerate environments for camera or LiDAR. In the future, the drone detection can be upgraded to a auxiliary-free method (e.g., no reflective tape), and the initialization can be more effective. Besides, the scale of the swarm can be larger, more drones will be produced to complete various tasks.

VI. ACKNOWLEDGEMENT

This work is supported in part by the University Grants Committee of Hong Kong General Research Fund under Project 17206421 and in part by DJI under the grant 200009538. The authors gratefully acknowledge Livox Technology for the equipment support and thank Ximin Lyu for the experimental site support during the whole work.
