

# Decentralized Visual-Inertial-UWB Fusion for Relative State Estimation of Aerial Swarm

Hao Xu, Luqi Wang, Yichen Zhang, Kejie Qiu, Shaojie Shen

**Abstract**—The collaboration of unmanned aerial vehicles (UAVs) has become a popular research topic for its practicability in multiple scenarios. The collaboration of multiple UAVs, which is also known as aerial swarm is a highly complex system, which still lacks a state-of-art decentralized relative state estimation method. In this paper, we present a novel fully decentralized visual-inertial-UWB fusion framework for relative state estimation and demonstrate the practicability by performing extensive aerial swarm flight experiments. The comparison result with ground truth data from the motion capture system shows the centimeter-level precision which outperforms all the Ultra-WideBand (UWB) and even vision based method. The system is not limited by the field of view (FoV) of the camera or Global Positioning System (GPS), meanwhile on account of its estimation consistency, we believe that the proposed relative state estimation framework has the potential to be prevalently adopted by aerial swarm applications in different scenarios in multiple scales.

## I. INTRODUCTION

The collaboration of unmanned aerial vehicles (UAVs) has become a more and more popular research topic in the recent years for its practicability in inspection, search & rescue, and light show performances. Aerial swarm is simply the collaboration of large amount of aerial vehicles. During the collaborative missions, it is crucial to perform real-time state estimation for each drone in order to plan smooth trajectories without collisions. Currently, the state estimations of aerial swarm are widely adopting either Global Positioning System (GPS) based methods [1] in outdoor environments or motion capture system and Ultra-WideBand (UWB) based methods [2] [3] [4] in indoor environments. However, the limitations of these methods are obvious: Due to the constraints of the GPS signal, state estimation for aerial swarm using GPS based methods can only be adopted in outdoor environments. Meanwhile, the system can only provide an accuracy at the magnitude of meters, which is not sufficient for tight formation flying. An alternative method is to increase the accuracy utilizing the Real-Time Kinematic positioning system [5]. Although the accuracy of centimeter-level positioning can be achieved by making use of the RTK-GPS framework, a ground station is compulsory to manipulate the system, causing the whole system to be clumsy and complicated to operate, which means that the framework can scarcely be adopted in an unknown environment. Despite that motion

All authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. {hxubc, lwangax, yzhangec, kqiuua}@connect.ust.hk, eeshaojie@ust.hk  
This work was supported by HKUST-DJI Joint Innovation Laboratory and HKUST institutional fund.



Fig. 1. Indoor aerial swarm formation flight with 5 drones. The customized drone platforms are circled in the snapshot.

capture or traditional UWB based localization systems in [2] [3] [4] can be functional in the indoor scenarios where the GPS signal is not available, the installation external cameras or ground anchors are still required, which is not suitable for the navigation in unknown environments.

In order to get rid of the bulky external sensors, so as to have the capability of performing swarm flying in unknown environments, adopting a visual-inertial state estimation method [6] can be a possible solution, which can provide effective and robust state estimations for UAVs in indoor and outdoor, known and unknown environments. Nevertheless, visual-inertial odometry (VIO) is only capable of providing the state estimations in local frames. Together with the drift issue, it is apparent that a simple one-time alignment of the coordinate systems of the individual UAVs before take-off is not sufficient during formation flying. It is intuitive to perform detection for other vehicles from the images captured by the on-board cameras so as to obtain relative state estimations [7]. Nonetheless, the estimation of the vehicles that are out of the field of view (FoV) of the on-board cameras can be a tricky problem. Hence, A sensor with omnidirectional sensing range and can be carried by a UAV, like a UWB module, is desired for the state estimation.

As a result, in this paper, we propose a vision-UWB based on-board real-time relative state estimation system which can be adopted in aerial swarm frameworks. The system is

- 1) Decentralized: Each vehicle perform relative state estimation individually and share information with others;
- 2) Flexible: Only on-board sensors and computing power, including stereo cameras, inertial measurement units

- (IMU), UWB modules, as well as DJI Manifold2-Gs are utilized. No external off-board sensors or computing power are required;
- 3) Robust: The system can handle the lack of the detection results as well as malfunctions of individual vehicles during swarm flights.

Our approach presents a comprehensive solution for on-board relative state estimation, eliminating the environmental restrictions of previous localization methods, which provides other researchers an opportunity to apply aerial swarm motion planning techniques in complex environments.

## II. RELATED WORKS

In recent years, it is shown that researches on aerial swarms are more likely to focus on planning and control, and the relative state estimation is always leaped by adopting external positioning systems, including GPS [1], motion capture system [8] or UWB modules with anchors [3], which are all centralized system, requiring bulky external devices. There is still no state-of-art solutions for decentralized relative state estimations, which can be adopted on aerial swarm applications. Up to present, there are several proposed solutions or systems but the practicability or the performance are generally not satisfactory. As mentioned previously in Sec. I, it is intuitive to use sensors like camera and UWB modules to perceive relative state estimation and the recent researches generally follows the two branches.

Several works have adopted the UWB based methods for relative localization. In [9] [10], the triangulation with distance measurements from the UWB modules are combined with the self displacement from optical flow method to obtain a relative state estimation. Only a 2-D translation is estimated and the initialization requires movement of a drone while other drones keep stationary, which limit the application scenario. In [11], another decentralized system using Extended Kalmen Filter to fuse the information from IMU and relative distance is introduced. The system is claimed to be the first attempt to realize a distributed localization framework with low cost sensors. Nevertheless, there are still large amount of challenges, including the precision problem preserved, making it difficult to put in practice.

A series of work on vision-based relative state estimation has been conducted by M. Saska and his team. In [12], a vision based framework is gradually developed. In the system, each drone is equipped with a circle marker for the detection purpose, which can be quite redundant. In [13] [14], additional UltraViolet Direction And Ranging (UVDAR) sensor and active marker are installed on the drone for tracking purposes. Nonetheless, all the systems may encounter potential failure or drift when the markers are out of the FoV of the cameras, which limits the application scenario to be leader-follower formation.

## III. RELATIVE STATE ESTIMATION FRAMEWORK

The complete relative state estimation structure is shown in Fig. 2. A combination of a detector and a tracker is adopted

for the motion capture of other drones when visible. Meanwhile, the measurements of relative distance from the UWB modules, together with the VIO are incorporated as well to optimize the 4 degrees of freedom (4DoF) relative state estimation, which consists of the three axes of translation  $x$ ,  $y$ ,  $z$ , and yaw rotation  $\psi$ .

We propose a two-stage optimization based relative state estimate for aerial swarm. The first stage is the fusion of visual and inertial data from stereo camera and IMU using VINS-Fusion [6]. The second stage is using the VIO result together with detection and UWB relative measurements for optimization-based fusion. The obtained relative state estimations are then sent back to facilitate the detection.

### A. Notations

In order to help to understand the estimation framework, the following notations are defined below and followed by the rest of the paper.

### NOMENCLATURE

$(\hat{\cdot})$	The estimated state.
$\mathbf{z}_i^t(\cdot)$	The measurement data at time $t$ .
$b_i(\cdot)$	State in drone $i$ 's body frame.
$V_i(\cdot)$	State in drone $i$ 's VIO frame.
$V_k \mathbf{P}_i^t$	Equals to $\begin{bmatrix} \mathbf{R}_z(V_k \psi_i^t) & V_k \mathbf{x}_i^t \\ 0 & 1 \end{bmatrix}$ . The pose of drone $i$ in drone $k$ 's VIO frame at time $t$ , which is from the result detecting and tracking. For simplification, the notation of $\mathbf{P}_i^t$ represents $V_i \mathbf{P}_i^t V_k \mathbf{R}_z(V_k \psi_i^t)$ represent the rotation matrix rotate over z axis with angle $V_k \psi_i^t$ .
$V_k \mathbf{x}_i^t$	Equals to $[V_k x_i^t, V_k y_i^t, V_k z_i^t]^T$ . The translation part of $V_k \mathbf{P}_i^t$ .
$\delta \mathbf{P}_i^t$	The transformation matrix from time $t - 1$ to $t$ of drone $i$ from the VIO result, i.e. $\mathbf{P}_i^t = \delta \mathbf{P}_i^t \mathbf{P}_i^{t-1}$ .
$d_{ij}^t$	Distance between drone $i$ and drone $j$ at time $t$ .

### B. Problem Formulation

For an aerial swarm system that contains  $N$  drones, the relative state estimation problem can be represent as: For an arbitrary drone  $k$ , estimate  $b_k \mathbf{P}_i^t$  for each drone  $i$  at time  $t$ .

The formulation for relative state estimator is

$${}^{b_k} \hat{\mathbf{P}}_i^t = \arg \min_{{}^{b_k} \hat{\mathbf{P}}_i^t} ({}^{b_k} \mathbf{P}_i^t)^{-1} ({}^{b_k} \hat{\mathbf{P}}_i^t). \quad (1)$$

Together with the ego motion state  $\hat{\mathbf{P}}_k^t$  estimated by the VIO, the pose of drone  $i$  in drone  $k$ 's VIO frame at time  $t$  can be easily retrieved:  $V_k \hat{\mathbf{P}}_i^t = \hat{\mathbf{P}}_k^t {}^{b_k} \hat{\mathbf{P}}_i^t$ .

### C. The First Stage: VIO

As mention above, the first stage of the framework is to utilize the image from the stereo camera and the IMU data to obtain the VIO. In this stage, a GPU accelerated version of VINS-Fusion<sup>1</sup> [15], an optimization based visual-inertial state estimator is adopted. The tightly coupled VIO result

<sup>1</sup><https://github.com/pjrambo/VINS-Fusion-gpu>

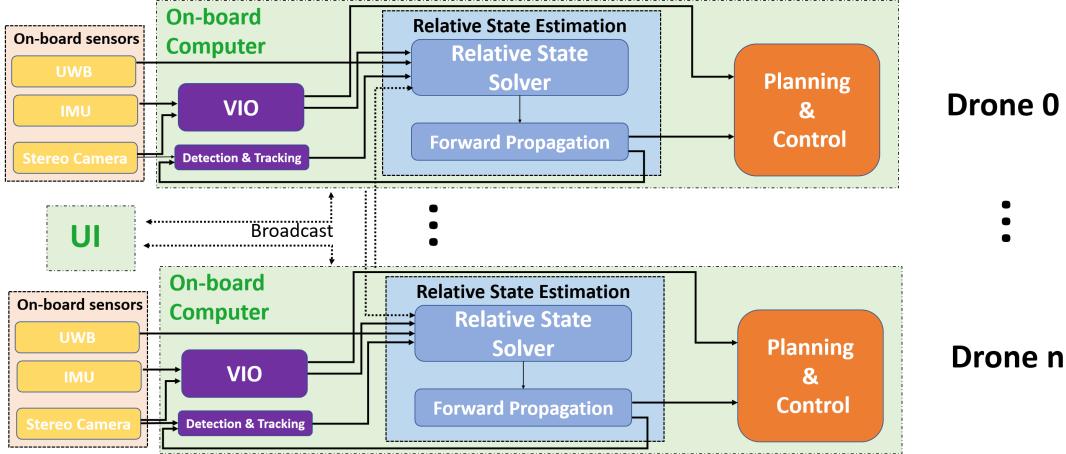


Fig. 2. The system architecture of the whole aerial swarm framework. The data from the on-board sensors are processed, then broadcast to all the other terminals. The relative state estimator on each of the on-board computer collects both on-board and broadcast information, including the relative distance from the UWB modules, the VIO, as well as the detection results, and perform optimization and prediction to obtain real-time relative state estimations. The estimation result is sent back to facilitate the matching procedure of detection and tracking meanwhile serve the planning and control. The ground station obtain the real-time information from the drones to monitor the flight status and concurrently send the commands to the drones. All the communications between the devices are through UWB broadcast.

is treated as one of the “measurements” to the second stage. From the VIO of individual drones, the transformation matrix measurements at time  $t$   $\mathbf{z}_{\delta \mathbf{P}_i}^t$  can be obtained and formulated with a Gaussian model:

$$\mathbf{z}_{\delta \mathbf{P}_i}^t = \delta \mathbf{P}_i^t + \mathcal{N}(0, \sigma_{vio}^2) \quad (2)$$

The equation does not contain the change of frame because  $\delta \mathbf{P}_i^t$  is frame invariant.

Meanwhile, the VIO is also adopted for planning and control purposes of the system.

#### D. Second Stage

Besides the VIO result from the first stage, the relative measurements from detection and tracking as well as distance are required for the optimization in the second stage.

1) *Detection and Tracking of Aerial Platforms:* For the detection of the aerial platforms, YOLOv3-tiny [16] is adopted, which is one of the state-of-art convolutional neural network (CNN) detectors, providing real-time object detection on the on-board computer. To utilized the detector for detecting our customized platforms, an additional training set captured by the on-board camera, as well as containing the drones are labeled and feed into the CNN. After the training, the network is able to efficaciously detect our aerial platforms. The detection results of 2-D bounding-boxes are combined with MOSSE trackers [17] to provide the tracking results with a higher frequency. Then, for each 2-D bounding-box from the tracking result, the depth of the center is retrieved from the depth camera. According to the pin-hole camera model, the 3-D relative position from the detection and tracking result can be recovered and the relative position in drone  $i$ 's body frame  ${}^{b_i} \mathbf{z}_{DT_j}^t$  can be retrieved according to the extrinsic parameters of the camera, where the footnote  $(\cdot)_{DT}$  denotes the estimation result from detection and tracking. Besides the position of a bounding-box, the size also provides information about the

depth. After the re-projection of a bounding-box onto the 3-D space, if the size immensely differs from the real size of the detected drone, the detection will be treated as an outlier. The detection and tracking result is also formulated with a Gaussian model:

$${}^{b_i} \mathbf{z}_{DT_j}^t = ({}^{V_k} \mathbf{P}_i^t)^{-1} {}^{V_k} \mathbf{P}_j^t + \mathcal{N}(0, \sigma_{det}^2) \quad (3)$$

where  $(\cdot)_T$  represents the translation in the transformation.

However, in our aerial swarm, there is another issue that all the platforms have a similar appearance, which cannot be distinguished by the detector and the tracker. Therefore, a matching algorithm is required for labeling the drones with their unique IDs. Once a drone is detected, the position from the visual measurement and the final estimation of the whole framework are compared and the matches are performed based on the principle of least disparities meanwhile should be less than a preset threshold value. Otherwise, the measurement will be treated as an outlier as well. After the drones are successfully labeled, once a new detection frame arrives, if the bounding-boxes of the tracker and the detector with the same ID have sufficient overlap, both the detection and the tracking are valid. Otherwise the matching procedure will be re-executed.

2) *UWB distance measurements:* From the UWB modules, the relative distances between each pair of the UWB nodes can be obtained, which are the UWB distance measurements  $\mathbf{z}_{d_{ij}}^t$ . The measurements are also modeled with a Gaussian noise:

$$\mathbf{z}_{d_{ij}}^t = \| {}^{V_k} \mathbf{x}_i^t - {}^{V_k} \mathbf{x}_j^t \|_2 + \mathcal{N}(0, \sigma_d^2) \quad (4)$$

3) *Data Frame, Keyframe Selection and Sliding Window:* All the above mentioned measurements and VIO are broadcast to all the terminals and a frame consists of the measurements and VIO at a specific time:

- The distance measurements of each two pair of drones  $\mathbf{z}_{d_{ij}}^t$ , provided by the UWB modules.

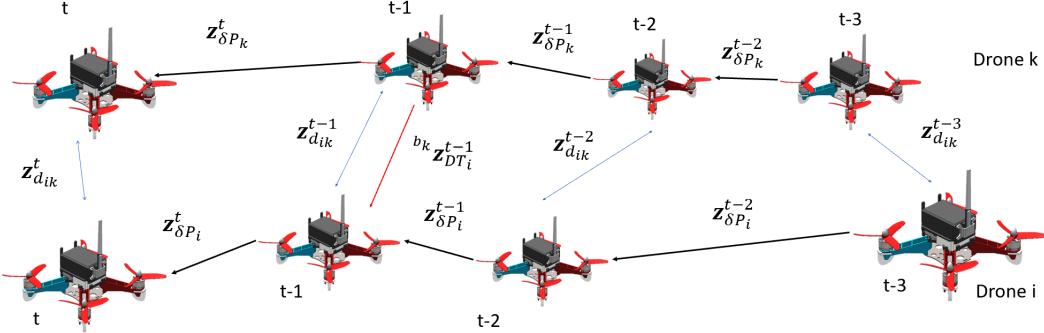


Fig. 3. An illustration of the sliding window relative state estimation. The notations are stated in Sec. III-A. The black arrows represent the transformation between the poses, which are provided VIO; the blue arrows indicate the relative distance measurements from the UWB modules; the red arrow depicts the relative 3-D translation from the result of detection and tracking.

- The tightly-coupled VIO results  $V_i \mathbf{z}_{\mathbf{P}_i}^t$  from the ego state estimation of individual aerial platforms using VINS-Fusion [15], which is treated as ‘‘VIO measurements’’.
- The relative translation measurements obtained from the detector and tracker  $b_i \mathbf{z}_{DT_j}^t$  stated in Sec.III-D.1.

For drone  $k$ , a frame is selected to be a keyframe if one of the following criteria is satisfied:

- The frame contain new drones.
- Drone  $k$  moves for a distance which is more than a certain threshold  $td_0$  compare to last keyframe.
- Drone  $k$  moves for a distance which is more than a certain threshold  $td_1$  compare to last keyframe and it detects or is detected by other drones.

Where  $td_0$  and  $td_1$  are adjustable parameters and typically set following the relation of  $0 < td_1 < td_0$ . A sequence of keyframes in chronological order forms a sliding window for optimization. An illustration of a sliding window is shown in Fig. 3. In practice, the maximum size of the sliding window for optimization is selected to be 50 to guarantee real-time and robust performance.

4) *Optimization-based Relative State Estimation:* With the sliding window formed previously in Sec. III-D.3, the optimization for the relative state estimation, which is the second stage can be performed. For a drone  $k$ , the full state vector  $\mathcal{X}_k$  for optimization is defined as:

$$\left[ \underbrace{\mathbf{V}_k \hat{\mathbf{P}}_0^{t_i T} \mathbf{V}_k \hat{\mathbf{P}}_0^{t_{i+1} T} \dots \mathbf{V}_k \hat{\mathbf{P}}_0^{t_{i+m-1} T}}_{m \times n \text{ elements}}, \underbrace{\mathbf{V}_k \hat{\mathbf{P}}_1^{t_i T} \dots \mathbf{V}_k \hat{\mathbf{P}}_{n-1}^{t_{i+m-1} T}}_{m \times n \text{ elements}} \right]^T$$

where  $n$  is the amount of drones in the swarm system,  $m$  is number of keyframes in the sliding window.

Instead of directly solving Eq.1, a bundle adjustment formulation is adopted for the optimization of the relative states. By minimizing the Mahalanobis norm of the residuals of the measurements, a maximum a posterior estimation can be obtained.

The optimization is expressed as the following formula-

tion:

$$\begin{aligned} \min_{\mathcal{X}_k} \left\{ \sum_{(i,j,t) \in \mathcal{L}} \left\| \mathbf{z}_{d_{ij}}^t - \left\| \mathbf{V}_k \hat{\mathbf{x}}_i^t - \mathbf{V}_k \hat{\mathbf{x}}_j^t \right\|_2 \right\|^2 \right. \\ \left. + \sum_{(i,j,t) \in \mathcal{D}} \left\| \left( (\mathbf{V}_k \hat{\mathbf{P}}_i^t)^{-1} \mathbf{V}_k \hat{\mathbf{P}}_j^t \right)_T - b_i \mathbf{z}_{DT_j}^t \right\|^2 \right. \\ \left. + \sum_{(i,t) \in \mathcal{S}} \left\| \left( \mathbf{z}_{\delta \mathbf{P}_i}^t \right)^{-1} \left( (\mathbf{V}_k \hat{\mathbf{P}}_i^{t-1})^{-1} \mathbf{V}_k \hat{\mathbf{P}}_i^t \right) \right\|^2 \right\} \end{aligned} \quad (5)$$

where  $\mathcal{L}$  is the set of all distance measurements,  $\mathcal{D}$  is the set of all detection measurements,  $\mathcal{S}$  is the set of all VIO ego motion measurements.  $\left\| \mathbf{z}_{d_{ij}}^t - \left\| \mathbf{V}_k \hat{\mathbf{x}}_i^t - \mathbf{V}_k \hat{\mathbf{x}}_j^t \right\|_2 \right\|^2$  is the residual of distance measurements;  $\left\| \left( (\mathbf{V}_k \hat{\mathbf{P}}_i^t)^{-1} \mathbf{V}_k \hat{\mathbf{P}}_j^t \right)_T - b_i \mathbf{z}_{DT_j}^t \right\|^2$  is the residual of detection and tracking measurements,  $(i,j)$  present the pair of successful detection and tracking of drone  $j$  detected by drone  $i$ ;  $\left\| \left( \mathbf{z}_{\delta \mathbf{P}_i}^t \right)^{-1} \left( (\mathbf{V}_k \hat{\mathbf{P}}_i^{t-1})^{-1} \mathbf{V}_k \hat{\mathbf{P}}_i^t \right) \right\|^2$  represents the residual of VIO results, ensuring the local consistency of the drone  $i$ 's trajectory measured in the two drones' VIO frames.

Simply solving Eq. 5 can directly provide the result of the relative state estimation with:

$$b_k \hat{\mathbf{P}}_i^t = (\hat{\mathbf{P}}_k^t)^{-1} \mathbf{V}_k \hat{\mathbf{P}}_i^t$$

The Ceres-solver [18], which is an open-source C++ library developed by Google for modeling and solving the Non-linear Least-Squares optimization problems is adopted, while the Trust Region with sparse normal Cholesky method is chosen. Meanwhile, since the planner or controller may require high frequency real-time poses, the real-time VIO from all the drones are composed with the optimized relative poses to provide a prediction at a higher rate of 100hz.

5) *Residual and Variable Pruning:* For an aerial swarm system containing  $n$  drones utilizing  $m$  keyframes in the sliding window, the optimization has  $4mn$  variables to solve. In practice, five drones and 100 keyframes are adopted in our experiment, resulting in solving 2000 unknowns, which is time-consuming and unnecessary. In the aerial swarm scenario, hovering of several vehicles can be a usual case.

In this situation, two residuals and the related variables can be pruned:

- If drone  $i, j$  are hovering, then for any other drone  $k$ , the distance residual mentioned previously in Sec. III-D.4 can be pruned, since the optimization is a formation of triangulation and the noise of distance measurements may cause the over-fitting issue.
- When a drone is hovering, the pose measurements should be fixed, meaning that the VIO residuals vanish.

After the pruning, the amount of residual elements is reduced by 64.3% and the number of variables is reduced by 50%, resulting in a speedup of 4 times for the optimization.

On a DJI Manifold 2-G on-board computer, in a scenario of aerial swarm with 5 drones, the average optimization time is reduced from 686.6ms to 169.6ms in a indoor flight case. Consider the drift of VIO can be negligible during such small peiod, our two-stage fusion algorithm can be treated as real-time.

*6) Estimator Initialization:* Since the optimization problem of estimating the relative pose mentioned in Sec. III-D.4 is highly nonlinear, directly fusing the measurements summarized Sec. III-D.3 without an appropriate initial value is almost impossible. Besides, due to the lack of observability, the relative yaw rotation cannot be solved unless the drones have performed enough translational movements. Hence in practice, the assumption that the observed drones are facing to close directions is made during the initialization, which means the yaw angle differences are considered to be small.

For drone  $k$  with sufficient movement, a few times of trial of optimization with random initial translational values are performed and the result with the smallest cost will be selected as the initialization result. Once the initialization has finished, the normal optimization will proceed with the obtained initial value.

#### IV. DETAILED SYSTEM IMPLEMENTATION

In order to verify the validity of the relative state estimation framework stated in Sec.III, a customized aerial swarm system consists of aerial platforms and a user interface (UI) is designed and implemented. All the state estimation and planning algorithms are running on the individual aerial platforms, while the ground station is only utilized for sending commands and monitoring flight status. The entire system is built under the ROS framework.

##### A. Aerial Platform

Five aerial platforms with identical hardware configurations are adopted for the swarm system. One of the platforms, which is a quad-rotor drone, is shown in Fig.4. The drone is equipped with a DJI N3 flight controller with an IMU embedded in, an Intel RealSense D435i stereo camera module, a Nooploop UWB module and a DJI Manifold2-G on-board computer. The raw data from the IMU at 400Hz and images from the stereo camera at 20Hz are fused together for the visual-inertial state estimation of individual drones using the GPU accelerated VINS-Fusion [15], while the images are also utilized for relative state estimation together with the

distance measurements from the UWB module. The attitude control of the drone is performed by the flight controller, while all other computations, including position control, state estimation and trajectory planning, are executed on the on-board computer. The communications between the drones as well as between the drones and the ground station are through the UWB modules, which are broadcasting data at 100Hz with synchronized time stamps.

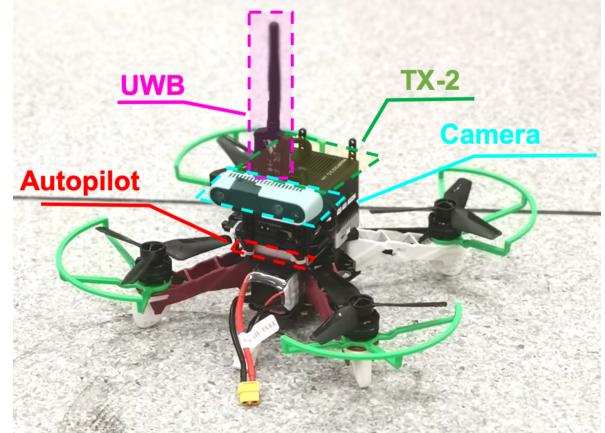


Fig. 4. One of the aerial platforms in the swarm system, which is equipped with a RealSense D435i depth camera, a DJI N3 flight controller, a Nooploop UWB module and a DJI Manifold2-G on-board computer with a Nvidia TX-2 chip.

## V. EXPERIMENT AND RESULT

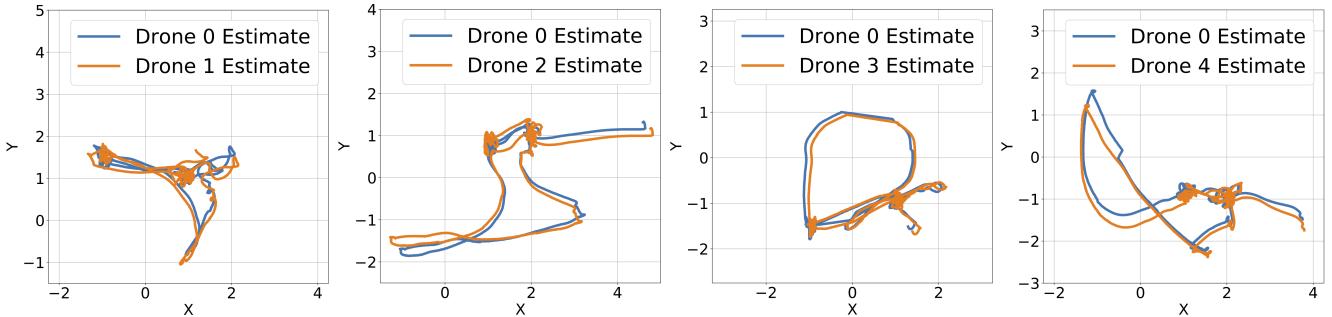
### A. Experiment Setup

The experiments are conducted in two indoor scenarios with the aerial swarm system mentioned in Sec. IV. The system consists of up to 5 drones is tested in two different indoor scenarios and compared with the ground truth data obtained from the OptiTrack motion capture system. A trajectory planner based on our estimated state for formation flying is introduced in the experiments to verify the validity and effectiveness of our relative state estimation method. The planner supports the transformation between 5 formations as well as translations of the swarm system. The traditional minimum-snap trajectory generator from way-points [19] is adopted, and the way-points are preset to avoid collisions during the changes of the formations.

### B. Experiment Result and Comparison

The relative state estimation framework is tested with 2, 3 and 5 aerial vehicles. Motion capture System is introduced as ground truth to ensure accuracy by comparing the relative poses from the motion capture system and our method.

Fig. 6 shows the estimated trajectory and the trajectory recorded by the motion capture system in a two-drone scenario while Tab. I and Tab. II show the error comparison between the proposed method and the ground truth. The UWB data and the detection data are abandoned in different scenarios to verify the effectiveness of the proposed sensor fusion method. According to Tab. I and II, it is shown that the RMSE error can reach 5cm level in 2 drones' aggressive flight and is similar with a triple-drone system. It should be noted that without the UWB measurements, it is extremely



(a) Relative positions of Drone 0&1 (b) Relative positions of Drone 0&2 (c) Relative positions of Drone 0&3 (d) Relative positions of Drone 0&4  
Fig. 5. The comparison of the real-time relative estimation between drone 0 and other drones in the formation flight with 5 drones. The relative estimations of drone 0 from other drone are inverted for comparison.

Ground Truth  
Estimate

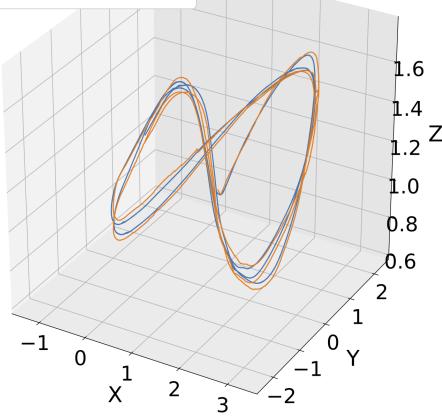


Fig. 6. The comparison of the observed trajectory of a drone between the relative estimation and the ground truth in two-drone scenario. The two frames are aligned for the visualization purpose. The RMSE of translation and yaw rotation are 6.84cm and 2.131 degree during the 34.553m flight.

difficult for the system to initialize with pure visual detection because of the difficulty in depth estimation and making distinctions between individual drones. Hence the benchmarked result without UWB is utilizing the initialization of the full system. For the five drones scenario, due to the experiment site limitation, the real-time relative state between drone 0 and other drones are provided in Fig. 5 to show the consistency of our proposed method.

When comparing with previous UWB based methods, the estimation system with ground anchors proposed in [3] reaches an RMSE error of 0.14m on the horizontal position and 0.28m on the total error, while [10] mentioned in Sec. II provides an error of 0.2m horizontally, without mentioning the vertical error, both of which are outperformed by the proposed decentralized relative state estimation, in terms of either completeness or precision. On the other hand, compared with the vision based system in [12], which provides 10 to 20cm estimation error, the proposed method provides better precision, meanwhile has the ability of handling the situation when the other vehicles are out of the FoV.

Furthermore, the consistency demonstrated in Fig. 5 shows the convenience of applying the method in different scales. Furthermore, the fully decentralized visual-inertial-UWB system is not limited by the FoV or GPS, which means that it has the potential to be prevalently adopted by aerial swarm applications in different environments.

TABLE I  
COMPARISON OF RELATIVE STATE ESTIMATION WITH GROUND TRUTH  
WITH A DOUBLE-DRONE SETTING

	<b>Proposed Method</b>	Without UWB	Without Detection
Traj. Length	34.553m		
RMSE of $b_k^t x_i^t$	<b>0.029m</b>	0.063m	0.038m
RMSE of $b_k^t x_i^t$	<b>0.058m</b>	0.076m	0.224m
RMSE of $b_k^t z_i^t$	<b>0.022m</b>	0.018m	0.046m
RMSE of $b_k^t \psi_i^t$	<b>2.131 deg</b>	3.665 deg	3.488 deg

TABLE II  
COMPARISON OF RELATIVE STATE ESTIMATION WITH GROUND TRUTH  
WITH A TRIPLE-DRONE SETTING

	<b>Proposed Method</b>	Without UWB	Without Detection
Traj. Length	16.382m		
RMSE of $b_k^t x_i^t$	<b>0.046m</b>	0.061m	0.072m
RMSE of $b_k^t x_i^t$	<b>0.052m</b>	0.075m	0.069m
RMSE of $b_k^t z_i^t$	<b>0.013m</b>	0.031m	0.176m
RMSE of $b_k^t \psi_i^t$	<b>1.015 deg</b>	2.106 deg	3.070 deg

## VI. CONCLUSION AND FUTURE WORK

In this paper, a novel decentralized visual-inertial-UWB relative state estimation framework is introduced. The proposed method combines detection, VIO and distance measurements together and provide the estimation result based on the optimization. The relative estimation framework is tested by extensive aerial swarm flight experiments with up to 5 customized drones and a ground station to demonstrate the feasibility and effectiveness. The relative estimation results are compared with ground truth data from the motion capture system, reaching a precision of centimeter-level, which outperforms all the Ultra-WideBand and vision based methods until present. On account of the flexibility and reliability of the system, this is the first fully decentralized relative state estimation framework that can be taken into practical applications. We believe that the proposed framework has the potential to be widely adopted by aerial swarms in different scenarios and in multiple scales.

With the proposed estimator, formation flights in various complex environments are no longer impossible. In the future, different control and planning algorithms for aerial swarm can be developed based on the relative estimation framework. We are planning to expand the system to be capable of performing safe flights in cluttered environments and even collaborative search & rescue tasks.

## REFERENCES

- [1] A. Jaimes, S. Kota, and J. Gomez, "An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles uav (s)," in *Proc. of the IEEE Intl. Conf. on Syst. of Syst. Eng. (SoSE)*. IEEE, 2008, pp. 1–6.
- [2] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors*, vol. 16, no. 5, p. 707, 2016.
- [3] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.(IROS)*. IEEE, 2015, pp. 3131–3137.
- [4] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*. IEEE, 2015, pp. 1730–1736.
- [5] S. Moon, Y. Choi, D. Kim, M. Seung, and H. Gong, "Outdoor swarm flight system based on rtk-gps," *Journal of KIISE*, vol. 43, no. 12, pp. 1315–1324, 2016.
- [6] 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.
- [7] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajinik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in gps-denied environments using onboard visual relative localization," *Auton. Robots*, vol. 41, no. 4, pp. 919–944, 2017.
- [8] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*. IEEE, 2017, pp. 3299–3304.
- [9] K. Guo, Z. Qiu, W. Meng, L. Xie, and R. Teo, "Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments," *Int. J. Micro Air Veh.*, vol. 9, no. 3, pp. 169–186, 2017.
- [10] K. Guo, X. Li, and L. Xie, "Ultra-wideband and odometry-based cooperative relative localization with application to multi-uav formation control," *IEEE Trans. Cybern.*, 2019.
- [11] S. Goel, "A distributed cooperative uav swarm localization system: Development and analysis," in *Proc. of the Intl. Tech. Meeting of The Satellite Division of the Inst. of Nav. (ION GNSS+)*, 2017, pp. 2501–2518.
- [12] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajinik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *Auton. Robots*, vol. 41, no. 4, pp. 919–944, 2017.
- [13] V. Walter, M. Saska, and A. Franchi, "Fast mutual relative localization of uavs using ultraviolet led markers," in *Proc. of the Intl. Conf. on Unm. Air. Syst. (ICUAS)*, 2018.
- [14] V. Walter, N. Staub, A. Franchi, and M. Saska, "UV-DAR System for Visual Relative Localization with application to Leader-Follower Formations of Multirotor UAVs," *IEEE RAL - online first*, 2019.
- [15] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," *arXiv*, 2019.
- [16] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.
- [17] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recg. (CVPR)*. IEEE, 2010, pp. 2544–2550.
- [18] S. Agarwal, K. Mierle, and Others, "Ceres solver," <http://ceres-solver.org>.
- [19] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*. IEEE, 2011, pp. 2520–2525.