Detection, Localization, and Tracking of Multiple MAVs With Panoramic Stereo Camera Networks

Ye Zheng, Canlun Zheng, Xiaoyu Zhang, Fei Chen, Zhang Chen, and Shiyu Zhao, *Member, IEEE*

***Abstract*— Malicious use of micro aerial vehicles (MAVs) has become a serious threat to public safety and personal privacy in recent years. Motivated by this problem, we propose a systematic approach to monitor the intrusion of malicious MAVs based on a novel type of panoramic stereo camera networks. Each sensing node of such a network consists of 16 lenses that can form a 360-degree panoramic vision system. The 16 lenses further form 8 pairs of stereo cameras that can directly localize aerial targets. The effective range for a sensing node localizing a MAV like DJI M300 could reach 80 meters, which is much farther than existing commercial stereo cameras. In terms of algorithms, we propose i) a novel visual MAV detection algorithm based primarily on motion features of MAVs, ii) an efficient stereo localization algorithm based on sparse feature points, and iii) robust multi-target tracking and trajectory fusion algorithm to fuse the observations of different sensing nodes. The effectiveness, robustness, and accuracy of the proposed algorithms together with the overall system have been verified by extensive experimental tests. To the best of our knowledge, this is the first systematic approach to detect, localize, and track unknown MAVs in the literature. Our approach provides a scalable solution to securely cover large areas of interest against malicious MAV intrusion.**

***Note to Practitioners*—Micro aerial vehicles (MAVs) have been widely used in many domains nowadays. However, they have also brought many safety problems. To monitor the intrusion of malicious MAVs, this paper proposes a novel type of panoramic stereo camera networks that can detect, localize, and track multiple MAVs simultaneously. Such a network consists of a number of sensing nodes and a central node. Each sensing node is able to detect, localize, and track multiple MAV targets. The role of the central node is to fuse the observations from multiple sensing nodes to generate more accurate trajectories of the MAV targets and in the meantime secure a large area in a coordinated way. This paper presents the details of the prototype of the system and the key algorithms therein.**

***Index Terms*— MAV detection, MAV localization, multi-target tracking.**

Manuscript received November 30, 2021; revised March 17, 2022; accepted May 14, 2022. This article was recommended for publication by Associate Editor L. Bascetta and Editor K. Saitou upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 61903308 and Grant 62073183, and in part by the Hangzhou Key Technology Research and Development Program under Grant 20200416A16. *(Corresponding author: Shiyu Zhao.)*

Ye Zheng is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China, and also with the School of Engineering, Westlake University, Hangzhou 310024, China (e-mail: zhengye@westlake.edu.cn).

Canlun Zheng, Fei Chen, and Shiyu Zhao are with the School of Engineering, Westlake University, Hangzhou 310024, China (e-mail: zhengcanlun@westlake.edu.cn; chenfei@westlake.edu.cn; zhaoshiyu@westlake.edu.cn).

Xiaoyu Zhang was with the School of Engineering, Westlake University, Hangzhou 310024, China. He is now with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China (e-mail: zhangxiaoyu@westlake.edu.cn).

Zhang Chen is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: cz_da@tsinghua.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2022.3176294.

Digital Object Identifier 10.1109/TASE.2022.3176294

I. Introduction

While micro aerial vehicles (MAVs) have been widely applied in many domains nowadays, they have also brought many safety problems, such as interference with the normal operation of airports. How to detect the intrusion of malicious MAVs has attracted increasing research attention recently [1]. MAV detection is also a key technology for many other tasks. For example, cooperative vision-based MAV swarming requires that each MAV must be able to detect their neighboring MAVs in real-time [2], [3]. Moreover, with the rapid development of aerial logistics by MAVs, mutual detection of MAVs for collision avoidance is also a demanding technology [4].

The current methods of detecting MAVs can be divided into two categories: active and passive. Active detection of MAVs mainly relies on radar [5]. Although radar technology is relatively mature, the effectiveness of radar will be compromised significantly for detecting low-altitude MAVs in complex urban environments. Passive detection of MAVs relies mainly on detecting remote control signals and acoustic or visual features of MAVs [6]. Although these detection methods are of low cost and more flexibility, they all face unique challenges. For example, the detection of remote control signals can be easily disturbed in complex electromagnetic environments such as urban centers. If a MAV flies autonomously without remote control, this method will fail. Acoustic detection is also susceptible to environmental interference, especially in urban areas. Visual detection is a promising method and is the focus of this work.

In recent years, the visual detection of MAVs has attracted increasing research attention in both academia and industry. However, visual detection still faces many technical challenges.

First, detection based on monocular vision is not robust to the appearance of MAVs and the environmental background. MAVs lack unique appearance features. The appearances of different MAVs may vary dramatically. The background in urban environments can be very complicated. We have proposed a comprehensive dataset of MAVs and evaluated the performance of eight mainstream deep learning algorithms [7].

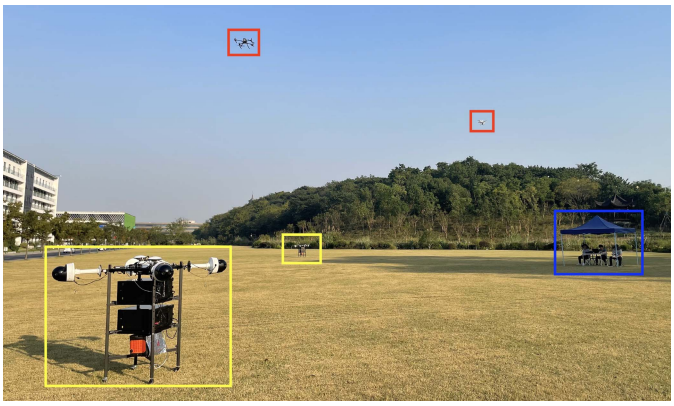


Fig. 1. The proposed panoramic stereo camera networks. In this specific experiment, we have two sensing nodes (yellow boxes) and one central node (blue box) on the ground to detect, localize and track the two MAVs (red boxes) in the air. The experimental video is available at *https://youtu.be/PNtRgkqgHw4*.

Although the algorithms show a certain generalization ability, how to detect unknown MAVs in complex environments remains largely unsolved.

Second, even if a MAV could be visually detected, its location is difficult to estimate accurately. That is because the size of the MAV is unknown and monocular vision is not able to recover the distance information. Although stereo vision can estimate depth information, its effective range is usually small (e.g., 15 m) due to a small baseline. Such a short effective range is not sufficient for detecting MAVs in large areas.

To overcome these challenges, we propose a novel type of panoramic stereo camera networks (see Fig. 1). In such a network, each sensing node consists of four cameras, each of which further has four lenses (see Fig. 2(a)). The 16 lenses of a sensing node form a 360° panoramic camera. Moreover, the 16 lenses form 8 pairs of stereo cameras (see Fig. 2(b)). Each pair of stereo cameras has a baseline of 1 m, which is much larger than the baseline of most commercial ones (typically less than 0.2 m). As a result, the effective range of depth estimation of the stereo cameras is much larger than commercial ones. Moreover, by fusing the measurements of multiple sensing nodes, the system could cover a much larger area. Such panoramic stereo camera networks can detect, localize, and track multiple MAVs simultaneously. To the best of our knowledge, this is the first work developing such type of camera networks for MAV detection. The novelties of the detection, localization, and tracking algorithms are summarized as follows.

1) Detection: The first step is to detect MAVs in the images captured by each lens. It is, however, still an open problem to visually detect unknown MAVs at present due to their unreliable appearance features. In our approach, we first prescreen potential MAV targets by detecting all moving objects using KNN [8] and then excluding the common objects that could be recognized by YOLOv5 [9]. The potential MAV targets may include a large number of false detections such as moving leaves or flags. We then refine the detection results by exploring the spatial and temporal features of MAVs so that MAVs could be robustly differentiated from interferences. The proposed approach is verified based on a dataset of MAVs collected in urban environments. This dataset contains a wide range of scenarios including three types of MAVs and a large number of disturbance sources such as persons, vehicles, trees, and flags. The experimental results verify the effectiveness of our proposed method under a variety of challenging conditions.

2) Localization: After MAVs have been detected in the images, stereo cameras could be used to localize them. Although stereo vision algorithms are mature, our system faces unique challenges. In particular, popular stereo techniques construct *dense* depth maps based on the pixel disparity. However, MAV targets are always sparse in the images. Constructing dense depth maps is not only unnecessary but also computationally impossible since we have 8 pairs of stereo cameras in one sensing node. To handle such a challenge, we propose a method to recover depth based on sparse image feature points. In particular, we extract the Oriented FAST and Rotated BRIEF (ORB) [10] feature points of detected MAV targets and then achieve robust feature matching across paired images. This method could localize multiple targets efficiently. Experimental results verify the efficiency and robustness of the proposed method. The effective localization range of MAVs like DJI M300 could reach 80 m with a relative error less than 10%.

3) Tracking: After MAV targets have been localized by each sensing node, it is important to fuse the observation of multiple sensing nodes to (i) correctly identify the total number of MAVs since one MAV may be observed by different sensing nodes, (ii) refine the trajectory smoothness since the detection of a MAV by a sensing node may be intermittent, and (iii) improve the localization accuracy. To that end, we propose a trajectory-based MAV tracking algorithm consisting of two parts. First, multi-target trajectory tracking is performed in each sensing node based on its local measurements. In this way, the measurement errors could be filtered to a certain extent. Then, each sensing node uploads the tracked trajectories to the central node for further fusion. The central node merges the trajectory segments into the fused trajectories based on the matching matrix by the Kuhn-Munkres algorithm [11]. Simulation and experimental results show that the proposed algorithm could track multiple targets effectively and robustly in the presence of various measurement errors.

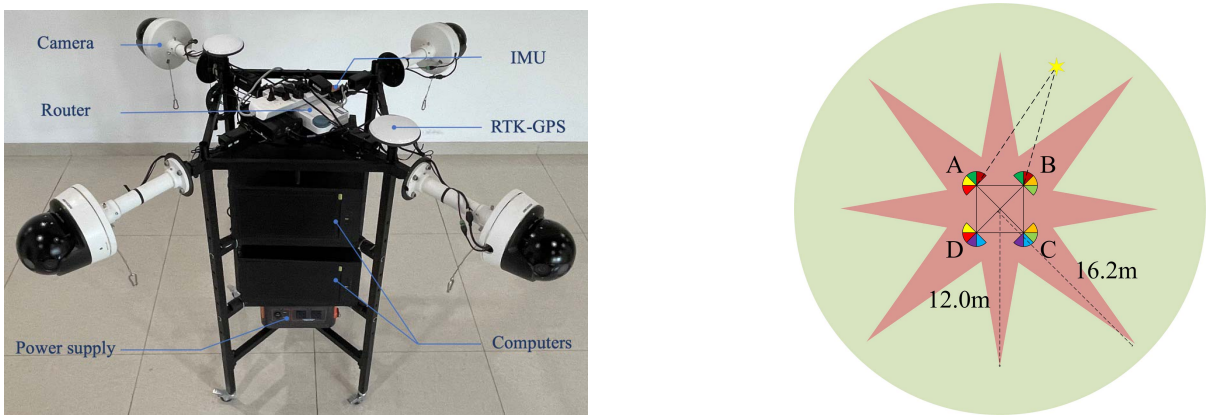
Finally, the effectiveness of the entire system consisting of the detection, localization, and tracking algorithms is verified by real outdoor experiments.

II. Related Work

This section gives a review of the existing studies of MAVs detection, localization, and tracking.

A. MAV Detection by Monocular Vision

The existing approaches in the field of MAV detection can be classified into two classes. The first is the conventional machine learning approach. In particular, the work in [12] detects MAVs with Harr-like feature-based AdaBoost. This



(a) Hardware platform of a sensing node

(b) Stereo camera structure of a sensing node

Fig. 2. Structure of a sensing node. In (a), the 4 lenses in each camera are parallel to the ground. The (b) demonstrates the configuration of stereo cameras in each sensing node and the coverage range of a sensing node. In (b), the baseline of each stereo vision is about 1 meter and the lenses with the same color are paired to form stereo cameras. The effective working area is illustrated in green, while the dead zone is illustrated in red.

method is verified to be effective in simple cases. In the work of [13], [14], the histogram of oriented gradients feature is adopted for training cascade detectors. Since multiple detectors are used, the computational burden is high. Motivated by moving object detection in the task of see-and-avoid, an optical flow method is developed in [15] and [16] to locate the moving targets, which are further recognized by template matching. This method would fail when the appearance of targets varies sharply. The work in [17] fuses the spatio-temporal context of the target for detection and achieves high performance in MAV detection. The work in [18] and [19] adopts event cameras and thermal cameras to detect and track moving objects, respectively.

The second is the model-free approach. In particular, the work in [7] evaluates eight state-of-the-art deep-learning-based algorithms for MAV detection. The experimental results show that the algorithms are not sufficient for real applications. To monitor the intrusion of MAVs, the work in [20] adopts a lightweight YOLOv3 and combines a wide-range camera and a zoomed camera as the sensing device to achieve high detection performance. The deep-learning-based object segmentation method has also been developed to detect MAVs. In particular, the work in [21] proposes a segmentation-based neural network, which employs spatio-temporal attention cues to achieve better performance than other algorithms. The work in [22] proposes a new MAV detector by modifying YOLOv2. The effective range for detecting MAVs like DJI M100 can reach 100 m in cluttered environments. Nevertheless, all the presented deep-learning-based approaches rely on the MAV images in the dataset. The generalization ability of the approaches is still limited.

B. Vision-Based MAV Localization

When a MAV has been recognized in an image, its relative bearing is trivial to calculate based on its pixel coordinate and the intrinsic camera parameters. Its relative distance is, however, challenging to recover especially using monocular vision. Although estimating depth from monocular images has received increasing research attention recently [23], it is an ill-posed problem and the methods are hard to be applied to different scenarios. Recovering target distance from bearing measurements has been studied extensively [24]–[26], but this approach requires the camera to observe the target from different angles. Some methods [27], [28] could estimate not only the distance but also the pose of the target. However, these methods rely on the saved models or templates. When the size of the target is unknown, it is still hard to estimate the distance accurately.

Stereo vision is a common method used to estimate target depth. In two-view geometry, the position of a 3D point can be triangulated from the matched points in stereo images [29]. In the work of [30], stereo vision is used to localize objects in indoor environments. To improve the localization performance for moving objects, the authors in [31] design a PTZ (Pan/Tilt/Zoom) stereo system. Although stereo vision is a well-developed technique, the commercial stereo cameras usually have a small baseline (e.g., 0.2 m) and hence a short effective range (e.g., 15 m). The relationship between the variance of range measurements and the baseline is analyzed in [32]. The work in [33] presents a pair of cameras to localize the 3D position of MAVs based on epipolar-like geometry. Some RGB-D devices (e.g., Kinect) can also provide accurate distance measurements. However, most of them only work in indoor environments, which is thus not suitable for our case.

C. MAV Multi-Target Tracking

One key problem in multi-target tracking is data association, which associates multiple observed targets' locations with the multiple estimated trajectories. To address the data association problem, the Hungarian method [34] is widely used. However, the one-step optimizing approach is not robust enough since the highest joint probability of the association is not always correct. Joint probabilistic data association (JPDA) [35] and multiple hypothesis tracking (MHT) [36] are the two popular algorithms for data association. There are some variants of the two algorithms. The work in [37] presents an approximation

version of JPDA called JPDAm, which uses the m-best solutions to an integer linear program. The work in [38] proposes an exclusive association sampling method to optimize the number of joint samples of JPDA. Through this method, the computational complexity of JPDA is reduced. Compared with JPDA, MHT is more suitable for finding the global optimal trajectory in multi-target tracking. The work in [39] utilizes online regularized least squares to achieve high efficiency in MHT. Since the above algorithms are used to associate the data obtained by one sensor, they are not suitable for the cases encountering data from multiple sensors.

Another important problem in multi-target tracking is trajectory refinement. To handle the observation errors, the work in [40] considers the trajectories of the targets are on the same plane. Thus, the problem of 3D trajectory tracking is simplified to 2D trajectory tracking. The work in [41] assumes the size of the MAV is known in advance. Then, the distance between the MAV and the camera can be obtained by the geometry method. Although these methods can reduce the observation error in MAV localization, they are not available in the case of tracking unknown MAVs.

In summary, the difference between our proposed approach and the existing ones is highlighted as follows. First, we propose a MAV detection algorithm based mainly on motion features, which are robust against the appearance change of MAVs. As a comparison, the other MAV detection algorithms are based mainly on appearance features, which may not be robust especially for detecting unknown MAVs. Second, our localization algorithm is to localize sparse MAVs based on feature extraction and epipolar geometry. As a comparison, the existing stereo matching algorithms are commonly for standard stereo configuration and these algorithms are to compute dense depth images, which is usually computationally expensive. Third, we propose a trajectory-based MAV tracking algorithm that incorporates Kalman tracking and the global nearest neighbor algorithm. To reduce the tracking error, a new trajectory refinement module is proposed.

III. System Overview

The proposed panoramic camera network consists of one central node and a number of sensing nodes (see Fig. 1). Each sensing node consists of 4 cameras, each of which has 4 lenses. As a result, each sensing node has 16 lenses (see Fig. 2(a)). The 16 lenses are paired to form 8 pairs of stereo cameras (see Fig. 2(b)). The baseline of each pair of stereo cameras is 1 m. The stereo vision forms a 360° panoramic vision system. Its effective localization range is demonstrated in Fig. 2(b). Other components of a sensing node include an inertial measurement unit (IMU), a real-time kinematic global positioning system (RTK-GPS), two computers, a router, and a movable power supply. The IMU and RTK-GPS are used to measure the accurate attitude and position of each sensing node in a common global coordinate frame. Each computer processes the images captured by 8 lenses. The router is used to connect the computers with the cameras and to transmit data with the central node.

The camera adopted in the sensing node is HIKVISION DS-2CD6984F-IHS/NFC, which is shown in Fig. 4. As can be seen, the 4 lenses in the camera are arranged one by one in a line. The specifications of the camera are given in Table I. To balance the computational efficiency and detection performance, the image resolution of each lens is set to 1280×720. Thus, the resolution of the stitched image processed by each computer is 1280×2880. Each pair of the stereo camera is well-calibrated in advance, by taking images of a checkboard and performing stereo calibration using OpenCV [42]. The field of view of each lens in the horizontal and vertical direction is 50° and 100°, respectively. In addition, the focal length of each lens is 2.8 mm and the frame rate is 25 FPS.

TABLE I
The Specification of the Camera Used in the Proposed System

Camera model (manufacturer)	DS-2CD6984F-IHS/NFC HIKVISION
Number of lenses	4
Resolution (height×width)	1280×720 (1 lens) 1280×2880 (4 lenses)
Focal length	2.8 mm
Field of view (each lens)	50° (horizontal) 100° (vertical)
Frame rate	25 FPS

The central node consists of a router and a ground control station computer. Its role is to fuse the measurements of multiple sensing nodes. While each sensing node could cover a limited area, multiple nodes could cover a much larger area. Since one MAV may be observed by multiple sensing nodes, the central node must be able to re-identify and smoothly fuse the trajectories of multiple targets observed by different nodes.

The workflow of the panoramic vision network for MAV detection, localization, and tracking is summarized as follows and illustrated in Fig. 3.

1) MAV targets are detected in each image captured by each lens. See Fig. 3(b) for illustration. The proposed MAV detection algorithm is mainly based on the motion features of the MAVs. This part of the work is detailed in Section IV.

2) After MAV targets have been detected in each image, pairs of images are used to localize the targets in their local coordinate frame. See Fig. 3(c) for illustration. This part of the work is detailed in Section V.

3) After MAV targets have been localized in each sensing node, the observations of different sensing nodes are fused in the central node by our proposed trajectory-based multi-target tracking algorithm. See Fig. 3(d) for illustration. This part of the work is detailed in Section VI.

Some remarks about the proposed system are given below. First, although a fisheye camera has the field of view more than 180°, it also has large image distortion, which introduces great challenges for MAV detection and localization. Second, although a PTZ camera can monitor a large area by continuously spinning, it is not able to detect, localize, and track multiple MAVs flying in different directions simultaneously. Third, there exist some blind spots of each sensing node (see Fig. 2(b)). However, since the proposed system is composed of multiple sensing nodes, if a MAV is in a blind spot of a sensing node, it can still be detected by other sensing nodes.

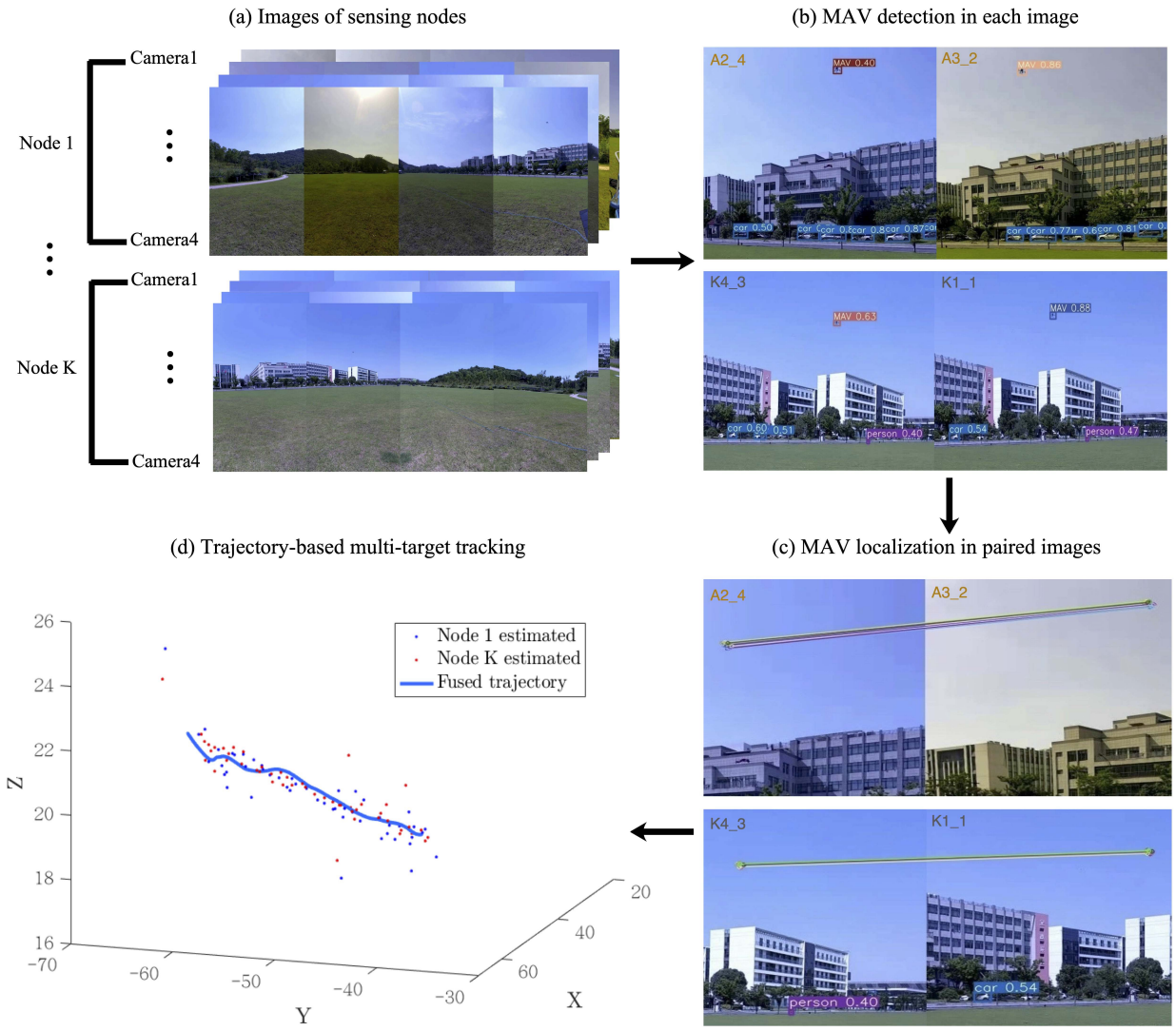


Fig. 3. The pipeline of the proposed system. In (a), there are a number of sensing nodes and each sensing node contains 16 images. The MAV in the images is detected in (b) and localized in (c). The spatial position of the target MAV is fused and tracked in (d). In (b) and (c), the image name labeled with the same color represents a pair of stereo cameras. The lines in (c) represent the process of feature matching.

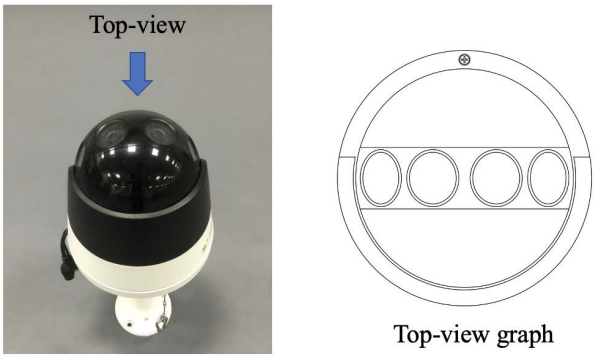


Fig. 4. Lens configuration of the camera used in the proposed system. This camera consists of 4 lenses. The right part of this figure is a top-view graph of the camera. It gives a more clear view of the lens configuration.

IV. MAV Detection by Each Lens

In this section, we propose a MAV detection algorithm to detect MAVs in the image sequences captured by each lens. Visual detection of unknown MAVs in general environments is still an open problem as reviewed in our recent work [7]. That is partially due to the fact that the appearance of different MAVs may vary vastly. Our proposed approach consists of two steps. The first is pre-screening potential MAV targets. The second is to refine the potential MAV targets based on the spatial and temporal properties of MAVs. The details are given as follows.

A. Pre-Screening Potential MAV Targets

The first step of our approach is to pre-screen potential MAV targets based on appearance and motion features. Since the appearance features of MAVs are not reliable, we use appearance features to exclude non-MAV targets instead of directly detecting MAVs. In particular, we use KNN to detect moving targets such as pedestrians, vehicles, and MAVs. Then, YOLOv5 is used to exclude the common objects such as pedestrians and vehicles from the moving objects. The rest

objects are regarded as potential MAV targets, which of course contain many false detections (i.e., moving but not MAVs) such as moving leaves or flags. These false detection will be further processed in Section IV-B.

The image sequence captured by each lens is processed in parallel by KNN and YOLOv5. The bounding boxes of moving objects are obtained by background modeling of KNN and refined by morphological processing. Since a structuring element with the size of 3×3 is used in morphological processing, the minimum pixel size of a MAV should be greater than 3×3 pixels. It is notable that the background such as cloud also varies slowly over time. As a result, we re-model the background every 75 frames (about 3 s). The inference results of YOLOv5 are processed by non-maximum suppression. Let $\mathcal{M} = \{\mathbf{m}_i\}_{i=1}^{p}$ denotes the bounding boxes of moving objects detected by KNN, and $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^{q}$ denotes the bounding boxes of common objects detected by YOLOv5. Each bounding box is represented by $[x_1, y_1, x_2, y_2]$, where (x_1, y_1) and (x_2, y_2) denote the up left corner and the lower right corner of a bounding box, respectively.

The maximum Intersection over Union (IoU) for each bounding box in $\mathcal{M}$ with all the bounding boxes in $\mathcal{C}$ is

$$\mathbf{g}_i = \max_{j\in\{1,2,\ldots,q\}}\left(\frac{|\mathbf{m}_i \cap \mathbf{c}_j|}{|\mathbf{m}_i \cup \mathbf{c}_j|}\right), \quad i = 1, \ldots, p. \tag{1}$$

Based on equation (1), we can obtain $\mathcal{G} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_p\}$. The bounding box $\mathbf{m}_i$ along with $\mathbf{g}_i$ greater than a threshold is regarded as a negative sample. The threshold is set to 0.5. By removing the negative samples from the moving objects, the candidates of flying MAV targets are

$$\mathcal{M}' = \mathcal{M} - (\mathcal{M} \cap \mathcal{C})_{IoU>0.5}. \tag{2}$$

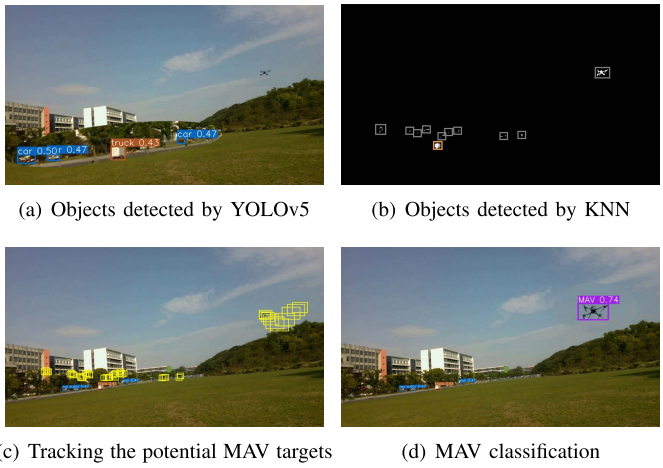
In equation (2), $\mathcal{M}' = \{\mathbf{m}'_i\}_{i=1}^{k}$.

The above pre-screening procedure may result in false detections such as moving trees and flags. We will remove these false detection based on their motion analysis. To do that, we adopt the multi-object tracking algorithm SORT [43] to track these detections in the image sequence to obtain their positions and sizes in each frame.

The processing results can be well demonstrated by the example in Fig. 5. On the one hand, the common objects detected by YOLOv5 as shown in Fig. 5(a) include three cars and one truck. The MAV is an unknown object for YOLOv5, so it is not detected. On the other hand, the moving objects segmented by KNN are shown in Fig. 5(b). It can be seen that many moving objects are detected such as the moving truck, shaking trees, and some other unknown but moving objects. The tracking results over a period of time are drawn in an image shown in Fig. 5(c). As can be seen, the moving trunk, which is detected by KNN, is removed from $\mathcal{M}$ since it is recognized as a non-MAV object by YOLOv5. Both the flying MAV and other targets such as shaking trees are all tracked.

B. Classification by Motion Analysis

In the previous step, potential MAVs have been detected and tracked. Next, a MAV classification algorithm based on motion analysis is proposed to distinguish MAVs from other



(a) Objects detected by YOLOv5 (b) Objects detected by KNN

(c) Tracking the potential MAV targets (d) MAV classification

Fig. 5. An illustrative example of the proposed MAV detection algorithm. In (a), (c), and (d), the blue box and red box represent "car" and "truck", respectively. The yellow boxes in (c) are the bounding boxes of the moving objects in previous frames. The purple box in (d) represents the detected MAV. Since the objects in the image are too small, the detection results in (a) and (d) are enlarged for a better demonstration.

interferences. In particular, the proposed classification model considers both temporal and spatial information, as defined below:

$$P = \beta P_S + (1-\beta)P_T, \tag{3}$$

where $P_S \in [0, 1]$ and $P_T \in [0, 1]$ denote the spatial metric and temporal metric of a MAV, respectively. In equation (3), $\beta \in [0, 1]$ is a trade-off parameter used to balance spatial and temporal metrics. The spatial metric P_S and temporal metric P_T represent the probability that a target can be classified as a MAV. These two metrics are defined based on a distance measure between two bounding boxes.

1) A Useful Distance Measure: To measure the distance between two bounding boxes, we adopt a metric recently proposed in [44]. This metric incorporates three geometric quantities (i.e. overlap area, central point distance, and aspect ratio) and hence comprehensively describes the relationship between two bounding boxes and even works for non-overlapping cases. In particular, the distance metric is

$$D(\mathbf{b}_i, \mathbf{b}_j) = 1 - IoU(\mathbf{b}_i, \mathbf{b}_j) + \frac{\rho^2(\mathbf{b}_i, \mathbf{b}_j)}{d^2(\mathbf{b}_i, \mathbf{b}_j)} + \alpha R(\mathbf{b}_i, \mathbf{b}_j), \tag{4}$$

where $\mathbf{b}_i$ and $\mathbf{b}_j$ are two bounding boxes obtained at time step i and j, respectively, $1 - IoU(\mathbf{b}_i, \mathbf{b}_j)$ denotes the non-overlapping area of the bounding boxes, $\rho^2(\mathbf{b}_i, \mathbf{b}_j)/d^2(\mathbf{b}_i, \mathbf{b}_j)$ denotes the normalized distance between the bounding boxes, $\rho(\mathbf{b}_i, \mathbf{b}_j)$ represents the central Euclidean distance between $\mathbf{b}_i$ and $\mathbf{b}_j$, $d(\mathbf{b}_i, \mathbf{b}_j)$ represents the diagonal length of the smallest enclosing box of $\mathbf{b}_i$ and $\mathbf{b}_j$, $\alpha R(\mathbf{b}_i, \mathbf{b}_j)$ measures the consistency of aspect ratio. The value of $d(\mathbf{b}_i, \mathbf{b}_j)$ can be computed by

$$d(\mathbf{b}_i, \mathbf{b}_j) = \sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2}, \tag{5}$$

where $x_{\min} = \min(x_1^i, x_1^j)$, $x_{\max} = \max(x_2^i, x_2^j)$, $y_{\min} = \max(y_1^i, y_1^j)$, $y_{\max} = \max(y_2^i, y_2^j)$. In equation (5), (x_1^i, y_1^i) and

Algorithm 1 Spatial Metric of a Potential MAV

Input: $\mathbf{m}_i'$, a bounding box of the potential MAV; $\mathcal{S}$, a set stored N bounding boxes of the potential MAV, $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\}$;
Output: The spatial metric of the target P_S;

DEFINITION $f(\cdot)$, a function counting the size of a set;
Put $\mathbf{m}_i'$ into $\mathcal{S}$ at the last position;
if $f(\mathcal{S}) = N$ **then**
  $D(\mathbf{s}_1, \mathbf{s}_N) = 1 - IoU(\mathbf{s}_1, \mathbf{s}_N) + \rho^2(\mathbf{s}_1, \mathbf{s}_N)/d^2(\mathbf{s}_1, \mathbf{s}_N) + \alpha R(\mathbf{s}_1, \mathbf{s}_N)$;
  $P_S = \begin{cases} 1, & \text{if } D(\mathbf{s}_1, \mathbf{s}_N) \geq 1 \\ kD(\mathbf{s}_1, \mathbf{s}_N), & \text{if } D(\mathbf{s}_1, \mathbf{s}_N) < 1 \end{cases}$;
  Remove the first element $\mathbf{s}_1$ from $\mathcal{S}$;
end
else
  $P_S = 0$;
end

(x_2^i, y_2^i) represent the up left corner and the lower right corner of the bounding box $\mathbf{b}_i$, respectively. Besides, the value of $R(\mathbf{b}_i, \mathbf{b}_j)$ can be computed by

$$R(\mathbf{b}_i, \mathbf{b}_j) = \frac{4}{\pi^2}\left(\arctan\frac{x_2^i - x_1^i}{y_2^i - y_1^i} - \arctan\frac{x_2^j - x_1^j}{y_2^j - y_1^j}\right). \quad (6)$$

Based on equation (6), the trade-off parameter α is designed as

$$\alpha = \begin{cases} 0, & \text{if } IoU(\mathbf{b}_i, \mathbf{b}_j) < \eta \\ \dfrac{R(\mathbf{b}_i, \mathbf{b}_j)}{1 - IoU(\mathbf{b}_i, \mathbf{b}_j) + R(\mathbf{b}_i, \mathbf{b}_j)}, & \text{if } IoU(\mathbf{b}_i, \mathbf{b}_j) \geq \eta. \end{cases} \quad (7)$$

In equation (7), η is set to 0.5.

2) Calculating Spatial Metric P_S: By observing the motion cues of MAVs and interferences in the video, we find that, in general, the interferences such as trees move much slower than MAVs, whose flying speed can be as high as 20 m/s. Thus, we give objects with high speed a high probability of being a MAV.

In particular, suppose that $\mathbf{b}_i$ and $\mathbf{b}_j$ are the two bounding boxes of the same target in two different images. The spatial metric characterizing the motion of the target between the two images is

$$P_S = \begin{cases} 1, & \text{if } D(\mathbf{b}_i, \mathbf{b}_j) \geq 1 \\ kD(\mathbf{b}_i, \mathbf{b}_j), & \text{if } D(\mathbf{b}_i, \mathbf{b}_j) < 1, \end{cases} \quad (8)$$

where k is a scale coefficient set to $2/3$. In equation (8), $D(\mathbf{b}_i, \mathbf{b}_j)$ denotes the distance between bounding box $\mathbf{b}_i$ and $\mathbf{b}_j$ based on the distance metric in equation (4).

It is noticed that, when a MAV is far away from the camera, its motion in the image may not be obvious over a short time even though its 3D speed is high. To address this problem, we calculate the spatial metric between two non-consequent images. The details of the algorithm calculating the spatial metric of each target are given in Algorithm 1. Through tracking the potential MAVs, the bounding box of each target can be obtained in each image frame. For each target, we define $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\}$ as the set of all the bounding boxes of the target over N image frames. The calculation of equation (8) is for $\mathbf{s}_1$ and $\mathbf{s}_N$.

Algorithm 2 Temporal Metric of a Potential MAV

Input: $\mathbf{m}_i'$, a bounding box of the potential MAV; $\mathcal{T}$, a set stored p bounding boxes by saving a bounding box every q frames, $\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_p\}$; T_s, the moving time of the target;
Output: The temporal metric of the target P_T;

DEFINITION $f(\cdot)$, a function that returns the size of a set;
if $f(\mathcal{T}) \leq p$ **then**
  for *each* $\mathbf{t}_j \in \mathcal{T}$ **do**
    $D_j(\mathbf{m}_i', \mathbf{t}_j) = 1 - IoU(\mathbf{m}_i', \mathbf{t}_j) + \rho^2(\mathbf{m}_i', \mathbf{t}_j)/d^2(\mathbf{m}_i', \mathbf{t}_j) + \alpha R(\mathbf{m}_i', \mathbf{t}_j)$;
  end
  if $D_{f(\mathcal{T})} \cdots D_2 D_1 \geq \xi$ **then**
    $T_s \leftarrow T_s + 1$;
  end
  if $i\%q = 0$ **then**
    Put $\mathbf{m}_i'$ into $\mathcal{T}$ at the last position;
  end
end
else
  Remove the first element $\mathbf{t}_1$ from $\mathcal{T}$;
end
$P_T = 1/(1 + e^{-\omega T_s + \lambda})$;

3) Calculating Temporal Metric P_T: We notice that the movement of MAVs is usually continuous and lasts for some time. As a comparison, the movement of interferences lasts usually for a short period of time. This phenomenon appears repeatedly for objects like fluttering flags and moving cars partially occluded by obstacles. Thus, we assign an object with a high probability of being a MAV if its motion is continuous and lasts for a long period of time. In particular, the temporal metric is defined as

$$P_T = \frac{1}{1 + e^{-\omega T_s + \lambda}}, \quad (9)$$

where T_s is the moving time. In equation (9), ω and λ are the regulatory factors that are set to $1/2$ and 20, respectively.

The moving time of the target T_s can be obtained by counting the number of consequent frames in which the target moves. To do that, we need to first define a moving indicator of a target:

$$s = \begin{cases} 1, & \text{if } \prod_{j=m}^{l} D_j(\mathbf{b}_i, \mathbf{b}_j) \geq \xi \\ 0, & \text{otherwise}, \end{cases} \quad (10)$$

where $s = 1$ and $s = 0$ indicate that the target is classified as moving or being static, respectively, by comparing the current bounding box $\mathbf{b}_i$ with the previous bounding boxes $\mathbf{b}_j$ with $j \in [m, l]$. Here, ξ is a threshold set to 1. This indicator is interpreted as follows. Since some objects such as the leaves of a tree move within a small area in images, their

bounding boxes overlap largely in a period of time. To remove such interferences, the indicator in equation (10) describes the distances from the bounding box $\mathbf{b}_i$ obtained in the current frame to those in the past l bounding boxes. If the indicator is less than the threshold ξ, suggesting that the target does not move significantly, the target is regarded as static.

Given the indicator s, we could further count the continuous moving time T_s. The details are given in Algorithm 2. In particular, we define a set $\mathcal{T} = \{\mathbf{t}_i\}_{i=1}^{p}$ for each target to store p bounding boxes, each bounding box is stored at a frame interval q. The p and q are set to 3 and 7, respectively. The distance between the coming bounding box $\mathbf{m}_i'$ and each bounding box in $\mathcal{T}$ is computed by equation (4). By taking all the distances into equation (10), we can obtain the current state of the potential MAV. In the end, equation (9) is adopted to compute the temporal metric of the potential MAV.

4) Classification: Once we have obtained P_S and P_T of a target, we can classify it based on equation (3). The trade-off parameter β in equation (3) is set to

$$\beta = \begin{cases} 0.3, & \text{if } T_s \leq \zeta \\ 0.5, & \text{if } T_s > \zeta. \end{cases} \tag{11}$$

In equation (11), ζ is a hyperparameter, and we set it to 20 frames. On the one hand, since the interferences are also moving, we set the weight of the spatial metric to be small at the beginning of tracking ($\leq$20 frames). It can eliminate the influence of the interferences. On the other hand, in order to reduce the dependence on temporal metric and avoid false detection, we reduce the weight of temporal metric after a period of time ($>$20 frames).

The result of classification is demonstrated in Fig. 5(d). As can be seen in this figure, the shaking trees and the discontinuous moving objects are removed, and the MAV is classified successfully among the tracked targets.

As a special case, a MAV may fly on a collision course to a sensing node. In this case, its position in the image plane hardly changes. However, the size of the target in the image plane still changes. When the target is close to the camera, its image size is large and hence results in a large value of the spatial metric. Then, the target can still be classified as a MAV.

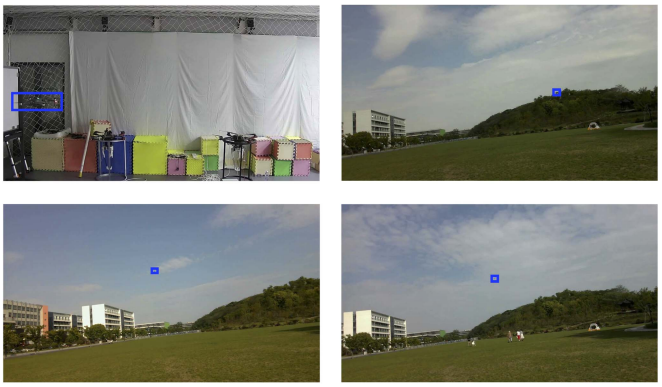
C. Evaluation of the MAV Detection Algorithm

To evaluate the performance of the proposed algorithm, we perform experiments on both single-lens images and panoramic images.

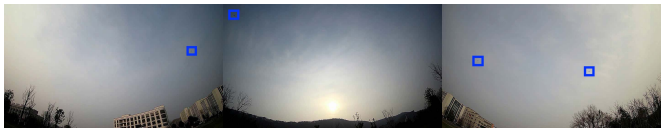
1) A MAV Dataset: First of all, we collect a dataset. This dataset contains three types of MAVs, which are DJI Phantom 4, DJI M300, and DJI Mavic 2. As shown in Fig. 6, the MAVs are different in terms of size and appearance. The dataset is composed of a single-lens dataset and a panoramic dataset.

The single-lens dataset contains image sequences captured in indoor and outdoor environments. The indoor images are captured under different illumination conditions. For the outdoor image sequences, we set the baseline illumination condition as the one for Mavic 2. The illumination condition of the Phantom 4 image sequences is weaker than the baseline, while the one of the M300 image sequences is slightly stronger. In addition, the image sequences of Phantom 4 and Mavic 2 are collected under cloudy weather condition, and the M300 image sequences are collected under sunny weather condition.

Fig. 6. Three MAVs in the dataset. From left to right, they are Mavic 2, Phantom 4 and M300, respectively. The indoor dataset contains Mavic 2, while the outdoor dataset contains all three MAVs.



(a) Samples in the single-lens dataset



(b) A sample in the panoramic dataset

Fig. 7. Image sample of our MAV dataset. The MAVs are labeled by blue bounding boxes. The single-lens dataset consists of the indoor dataset and the outdoor dataset. It has only one MAV in an image. While in the panoramic dataset, there is at least one MAV in each image. The images in the panoramic dataset are stitched by three sub-images, which are captured by a panoramic camera at the same timestamp.

The panoramic dataset is obtained by a panoramic camera with three lenses. We stitch the images captured by the three lenses to form a panoramic image. The panoramic dataset is taken from the urban environment with two M300 in a low light condition.

The samples of the dataset are shown in Fig. 7. It can be seen in this figure that the MAVs are of small size compared to the image size. Besides, the samples collected in an outdoor environment have many challenging interferences, such as fast-flying birds, pedestrians, moving cars, and shaking trees. The distance between the MAV and the camera is estimated based on the pin-hole camera model.

TABLE II
THE PERFORMANCE OF THE PROPOSED ALGORITHM UNDER DIFFERENT ILLUMINATION CONDITIONS IN THE INDOOR ENVIRONMENT

Image sequences	Illumination condition	Sequence length	Range (m)			Precision	Recall	F1-score	AP
			min	max	mean				
ID_IS1	285 LUX	251	4.06	9.14	5.36	99.20%	98.80%	98.40%	98.01%
ID_IS2	196 LUX	851	4.74	10.60	7.09	94.57%	81.90%	87.78%	77.77%
ID_IS3	65 LUX	951	5.32	12.21	6.78	79.77%	85.80%	82.68%	74.43%
ID_IS4	7 LUX	1046	3.84	12.43	5.75	84.22%	62.51%	71.76%	57.75%

TABLE III
THE PERFORMANCE OF THE PROPOSED ALGORITHM ON PANORAMIC IMAGES IN THE OUTDOOR ENVIRONMENT

Image sequences	MAVs	Sequence length	Range (m)			Precision	Recall	F1-score	AP
			min	max	mean				
OD_PIS1	M300	1000	29.81	88.18	48.06	99.72%	91.05%	95.19%	91.01%
OD_PIS2	M300	1351	30.57	91.07	49.39	100.00%	88.90%	94.12%	88.90%
OD_PIS3	2×M300	2162	31.40	230.92	57.85	99.86%	89.30%	94.29%	89.25%

TABLE IV
THE PERFORMANCE OF THE PROPOSED ALGORITHM FOR DIFFERENT TYPES OF MAVS IN THE OUTDOOR ENVIRONMENT

MAVs	Image sequences	Weather condition	Illumination condition	Sequence length	Range (m)			Precision	Recall	F1-score	AP
					min	max	mean				
Phantom 4	OD_IS1	Cloudy	Weak	1200	24.04	109.84	45.22	97.67%	94.42%	96.02%	93.36%
	OD_IS2			800	17.75	42.70	28.11	100.00%	96.00%	97.96%	96.00%
M300	OD_IS3	Sunny	Strong	1352	15.18	93.85	36.24	72.72%	98.96%	83.83%	96.48%
	OD_IS4			1491	13.31	146.04	42.93	100.00%	98.59%	99.29%	98.59%
Mavic 2	OD_IS5	Cloudy	Medium	948	8.99	65.73	22.50	100.00%	99.79%	99.89%	99.79%
	OD_IS6			1041	8.28	51.96	19.70	91.21%	99.71%	95.27%	99.48%

2) Evaluation Metrics: To verify the performance of the proposed algorithm, we evaluate it with Precision, Recall, F1-score, and Average Precision (AP).

In our experiment, if the MAV is successfully detected, we will regard the predicted bounding box as true positive (TP). Otherwise, it will be regarded as a false positive (FP). While if the MAV is undetected or mistaken for other objects, then the target will be regarded as a false negative (FN).

Precision reflects the proportion of real positive samples in the predicted positive samples. It is defined as Precision $= TP/(TP + FP)$. Recall is also an important factor for evaluating missing detection. It is defined as Recall $= TP/(TP + FN)$.

To comprehensively evaluate the missing detection and false detection, F1-score and AP are also adopted. F1-score is the harmonic mean of Precision and Recall, which is defined as F1-score $= 2 \times \text{Precision} \times \text{Recall}/(\text{Precision} + \text{Recall})$. AP is the area under the Precision-Recall curve. The area under the curve can be computed by $\text{AP} = \sum_{r=0}^{1}(r_{n+1} - r_n) \max_{\widetilde{r}:\widetilde{r} \geq r_{n+1}} \rho(\widetilde{r})$, where $\rho(\widetilde{r})$ is the Precision corresponding to the Recall at $\widetilde{r}$.

3) Evaluation Results: The testing results of the MAV detection algorithm on the indoor image sequences are shown in Table II. In ID_IS1, the MAV flies in front of the white curtain and is accompanied by an appropriate illumination condition. The detection algorithm has the best result in this case.

Table III shows the performance of the MAV detection algorithm on the panoramic images. Since the noises are filtered successfully by the proposed algorithm, a high Precision ($> 99\%$) can be obtained in the tests. In addition, increasing the number of MAVs has a limited impact on the computational performance of the algorithm.

The testing results on the image sequences with three different MAVs are listed in Table IV. In terms of Recall ($> 94\%$), the proposed algorithm has a low missing detection rate. In some cases with fewer interferences, the MAV detection algorithm has high Precision. However, due to partial occlusion by obstacles, some moving cars can not be recognized by YOLOv5, which results in a low Precision in OD_IS3.

From the detection results of M300 in Table III and Table IV, we can see that the algorithm can still generate good detection performance when the average distance between the camera and the MAV increases.

The illumination condition has a great impact on MAV detection. As can be seen in Table II, when the illumination condition becomes worse, although ID_IS4 has a smaller

TABLE V
THE PERFORMANCE OF DIFFERENT ALGORITHMS IN DETECTING UNKNOWN MAV

Algorithm	Cascade R-CNN	Grid R-CNN	RetinaNet	YOLOv5-x	Ours
AP	92.6%	92.2%	89.5%	91.7%	**97.7%**
Runtime	217 ms	313 ms	154 ms	**105 ms**	119 ms

average range than ID_IS2 and ID_IS3, the performance of the MAV detection algorithm still drops sharply. In addition, it can be seen from the results of Phantom 4 and Mavic 2 in Table IV that, the worse the illumination condition, the lower the performance of the proposed MAV detection algorithm.

We compare our algorithm with the state-of-the-art deep-learning-based object detection algorithms including Cascade R-CNN [45], Grid R-CNN [46], RetinaNet [47], and YOLOv5-x [9]. The algorithms are trained with image sequences of Phantom 4 and Mavic 2, and tested with images sequences of M300. Since the proposed MAV detection algorithm is not a data-driven algorithm, we can directly test it. As can be seen in Table V, the proposed MAV detection algorithm has the highest AP with a relatively low runtime.

In summary, the proposed MAV detection algorithm shows a stable and high performance in the MAV dataset. Although there are a few false detections, the MAVs are rarely missed. Since the MAV detection algorithm is based on an assumption that the MAV is a moving object, it would fail if the target hovers statically. In this case, the multi-target tracking module can still track it based on the information obtained in the last few steps. More details can be seen in Section VI.

V. MAV LOCALIZATION BY PAIRED STEREO CAMERAS

With the algorithm presented in the previous section, each of the 16 lenses can detect MAV targets. In this section, we show how to localize the detected MAV targets by paired cameras. Since the targets are commonly sparse in the images, our method only computes the 3D positions of the feature points belonging to the bounding boxes of these targets. Such a way is more efficient and robust than recovering dense depth maps using stereo cameras.

The pipeline of the proposed method, illustrated in Fig. 8, consists of two steps. The first step is data association, which is to match the multiple MAV targets detected on two paired images. The second step is matching and culling of feature points belonging to the MAV bounding boxes. The average position of the feature points is regarded as the final estimated position of a MAV.

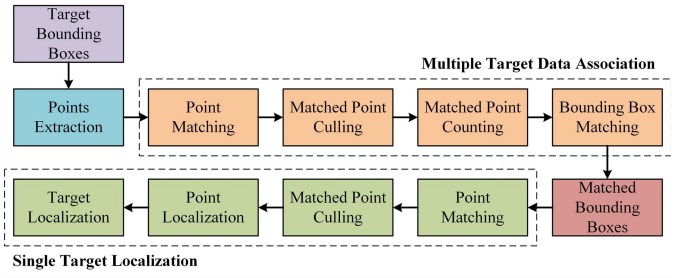
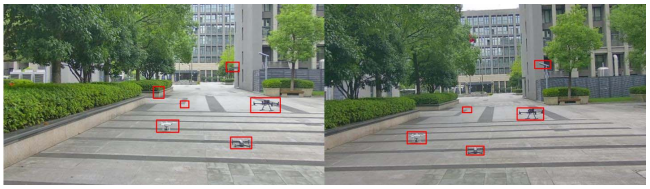
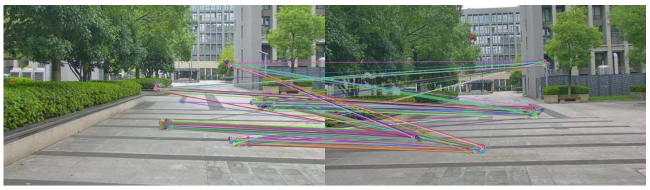


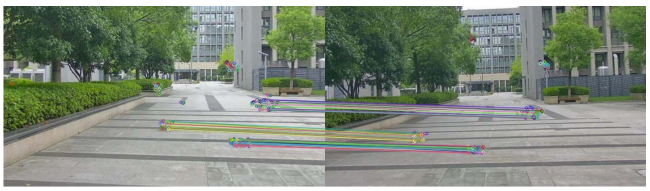
Fig. 8. The pipeline of MAV localization. Multiple MAV targets detected on two paired images are firstly matched based on feature matching. For matched targets, the features are matched and culled again. Finally, the 3D positions are computed.



(a) Targets are labeled by red bounding boxes.



(b) Point matching in left and right images. The matched points are connected by colorful lines, and some mismatched points exist.



(c) Point matching in left and right images. Those mismatched points are culled out based on the epipolar constraint.

Fig. 9. Multiple targets data association.

A. Bounding Box Matching

For two paired images, since multiple MAVs are detected in each of them, the first step is to match the bounding boxes in the left image to those in the right one. We assume that the bounding boxes sharing the most matched feature points correspond to the same MAV.

The procedure of bounding box matching is detailed as follows and illustrated by the experimental example in Fig. 9. First, as shown in Fig. 9(a), there are multiple bounding boxes in the left and right images. We also deliberately add some extra bounding boxes corresponding to no MAVs in both images to examine the robustness of our algorithm. The ORB features are extracted from all these bounding boxes. Second, the ORB features in the left and right images are matched based on their descriptors. The reason that ORB features are used is that they are fast to extract and match, and also robust to illumination and viewing direction changes. As shown in Fig. 9(b), the matched features, connected by colorful lines,

TABLE VI

LOCALIZATION ERROR. *Num* DENOTES THE NUMBER OF MATCHED FEATURE POINTS TO COMPUTE THE TARGET DISTANCE

Truth (m)	DJI M300			DJI Mavic 2			DJI Phantom 4		
	Num	Estimated (m)	Error (m)	Num	Estimated (m)	Error (m)	Num	Estimated (m)	Error (m)
15.0	73	13.83	1.17 (7.80%)	9	14.58	0.42 (2.80%)	7	15.19	0.19 (1.27%)
20.3	15	18.63	1.67 (8.23%)	5	18.73	1.57 (7.73%)	5	19.47	0.83 (4.09%)
25.2	18	23.94	1.26 (5.00%)	2	22.10	3.10 (12.3%)	2	24.25	0.95 (3.77%)
30.6	14	29.88	0.72 (2.35%)	4	29.70	0.90 (2.94%)	2	29.86	0.74 (2.42%)
35.6	8	34.18	1.42 (3.99%)	3	36.68	1.08 (3.03%)	2	34.77	0.83 (2.33%)
40.9	9	39.47	1.43 (3.50%)	2	42.49	1.59 (3.89%)	2	40.52	0.38 (0.93%)
45.5	9	44.39	1.11 (2.44%)	2	44.88	0.62 (1.36%)	1	45.42	0.08 (0.18%)
50.8	5	51.75	0.95 (1.87%)	2	49.38	1.42 (2.80%)	1	48.49	2.31 (4.55%)
55.5	4	56.43	0.93 (1.68%)	-	-	-	-	-	-
60.5	4	60.79	0.29 (0.48%)	-	-	-	-	-	-
65.9	0	**60.51**	**5.39 (8.18%)**	-	-	-	-	-	-
69.7	2	72.63	2.93 (4.20%)	-	-	-	-	-	-
74.6	0	**70.01**	**4.59 (6.15%)**	-	-	-	-	-	-
79.7	0	**79.18**	**0.52 (0.65%)**	-	-	-	-	-	-

suffer from severe mismatching. Third, mismatched points are culled based on stereo geometry. In particular, the extrinsic parameters are acquired by stereo calibration in advance. Thus, the corresponding epipolar line in the right image for every point in the left image can be computed. The matched points of the feature points from the left image should be on their corresponding epipolar lines in the right image, namely the epipolar constraint [29]. The system culls all matched points that violate the epipolar constraint. As shown in Fig. 9(c), the mismatched features are culled out.

B. Target Localization

Next, we compute the 3D position of each MAV based on the matched bounding boxes.

The procedure is detailed as follows and illustrated by the example in Fig. 10. First, we redo feature point extraction and matching. That is because, in the bounding box matching step, although the bounding boxes are matched successfully, only a few matched feature points are left. To enrich feature points inside each bounding box, we redo ORB feature extraction and matching. This time we only match features within matched bounding boxes. The point matching results are illustrated in Fig. 10.

Second, the 3D positions of all of the matched points in the bounding box can be computed based on stereo geometry [29]. However, it is notable that some feature points may belong to the background instead of the target MAV. Therefore, we must filter out these background points. The idea is that the 3D position (especially depth) of a background feature vastly differs from that of a flying MAV. Since the position of each MAV target is tracked over consequent time steps, if the 3D position of a feature differs from the 3D position of the MAV in the last time step significantly, then the feature is regarded as a background feature and filtered out. The rest features are regarded as the features belonging to the MAV. In the end, the target position is calculated as the average of 3D positions of the rest feature points.

C. Evaluation of the Proposed Method

To evaluate the effectiveness of the proposed localization algorithm, we conduct a series of experiments, in which three

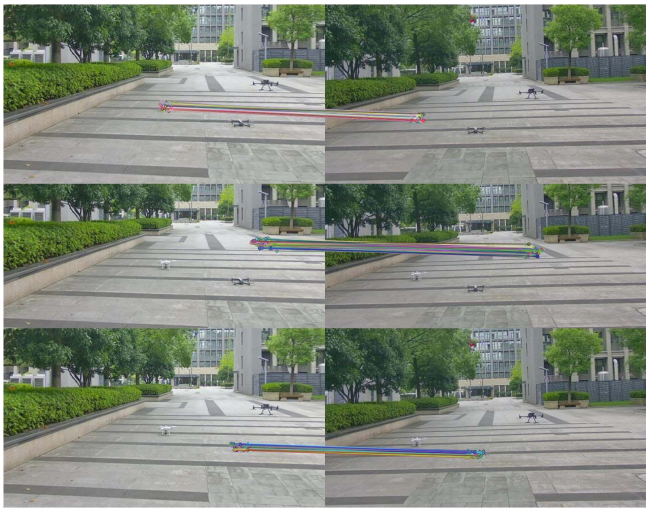


Fig. 10. Point matching in matched bounding boxes. Matched feature points in the left and right images are connected by lines.

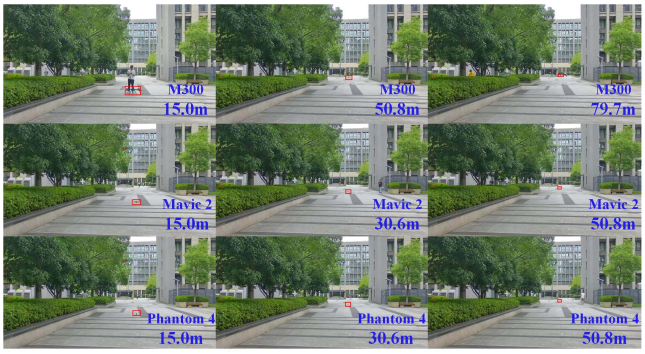


Fig. 11. Images of MAV localization experiments. The target MAVs from different distances are labeled by red bounding boxes.

MAVs are located at different positions (see Fig. 11). The three MAVs are DJI M300, Mavic 2, and Phantom 4, which are different in terms of both size and appearance. In the experiments, we measure the true distance of the MAVs using a laser range finder.

The experimental results are given in Table VI. First, for Mavic 2 and Phantom 4, whose sizes are relatively small, they

can be effectively localized within about 50 m. For M300, which is relatively large, the effective localization range is about 80 m, which is much larger than conventional stereo vision systems with small baselines. Second, the number of feature points that can be extracted and matched in the bounding boxes decreases as the MAV targets move far away from the camera. In the extreme cases where the bounding boxes are too small, no feature points could be extracted at all. For example, when M300 is placed at 65.9 m, 74.6 m, and 79.7 m, the feature points cannot be matched because the target is very small. In these extreme cases, we could simply treat the center of a bounding box as a feature point and then do matching and localization. Third, the relative localization error of our stereo system is around 10% in the worst case and normally much lower than 10%. The localization error is affected by a number of factors including the feature numbers, feature matching accuracy, and stereo calibration accuracy. The resolution of the images used in the experiments is 1280×720. If using images with higher resolution, the stereo system can localize farther MAVs and get more accurate results. Moreover, the localization accuracy can be further improved by fusing observations from multiple sensing nodes as shown in the following section.

VI. Data Fusion and Multi-Target Tracking

With the algorithms presented in the previous sections, each sensing node could detect and localize MAV targets in their local coordinate frames. The localized targets of each sensing node are uncorrelated either temporally or spatially. In this section, we present algorithms to identify how many MAVs exist and provide smooth trajectories of these MAVs.

To do that, we need to overcome some challenges. For example, the detection of a MAV by a sensing node may be intermittent, which may cause wrong temporal data association and hence wrong MAV trajectories. Moreover, the target localization of each sensing node must be converted to a common global reference frame based on the onboard IMU and GPS. Such conversion, however, may cause large localization errors due to the IMU and GPS measurement errors. The large localization error may severely compromise the performance of the data association and hence trajectory fusion among multiple sensing nodes.

To overcome these challenges, we propose a trajectory-based MAV tracking algorithm. Given the uncorrelated observation of multiple sensing nodes, this algorithm can return the total number of MAVs and track their positions. To reduce the tracking error, we propose a trajectory refinement module, which would also make the estimated trajectory smoother. The proposed algorithm consists of two parts (see Fig. 12). The first part is to track multiple MAVs in a single node. The second part is to fuse the trajectories provided by multiple sensing nodes.

A. Multi-MAV Tracking by a Sensing Node

In this part, each MAV is assigned with a Kalman tracker based on the constant velocity model for state estimation. When the new observed positions are available, they will be associated with the positions predicted by trackers, respectively.

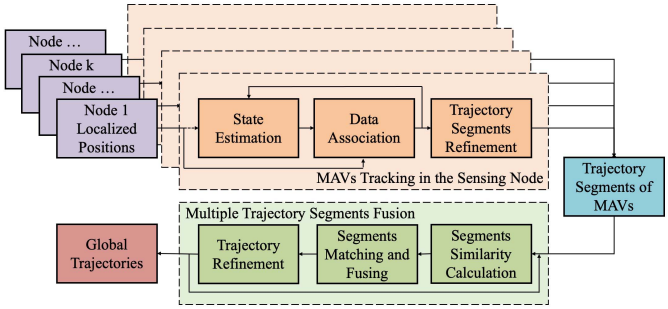


Fig. 12. The framework of trajectory-based MAV tracking algorithm. The dotted line in the module of the sensing node represents the initialization for the state estimation algorithm.

The Kalman tracker for each MAV is described as follows. Since the control input of the observed MAV is unknown, the state equation and measurement equation of the Kalman tracker are assumed to be

$$\begin{aligned} \mathbf{x}_t &= \mathbf{A}\mathbf{x}_{t-1} + \mathbf{w}, \\ \mathbf{z}_t &= \mathbf{H}\mathbf{x}_t + \mathbf{v}, \end{aligned} \tag{12}$$

where $\mathbf{x}$ is the state vector $\mathbf{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z})^\top$. In equation (12), (x, y, z) and $(\dot{x}, \dot{y}, \dot{z})$ are the position and velocity of the MAV, respectively. The measurement vector is the estimated position of the MAV. Here, $\mathbf{w}$ and $\mathbf{v}$ are the process noise and measurement noise, respectively. The state matrix $\mathbf{A}$ and the observed matrix $\mathbf{H}$ are

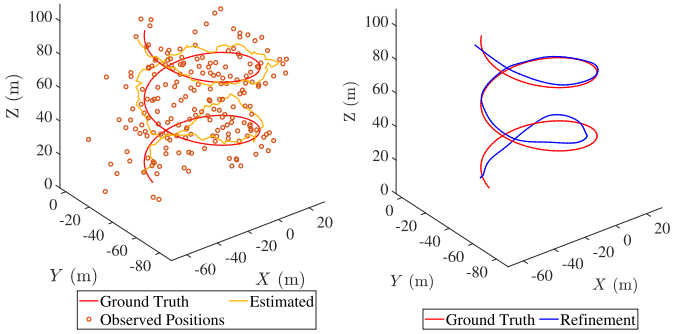
$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3}\delta t \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{I}_{3\times3} \mid \mathbf{0}_{3\times3} \end{bmatrix},$$

where δt is the sampling interval.

Data association is to match the observed MAV positions with the multiple Kalman trackers. Here, data association is achieved by the global nearest neighbor (GNN) algorithm, which has higher efficiency and lower error compared to other algorithms such as MHT and JPDA according to our experimental tests. The details of the data association process are given below. Let $\mathbf{Z}(t) = \{\mathbf{z}_i(t)\}_{i=1}^N$ be the set of N observed MAV positions of one node at time t, and $\mathbf{P}(t|t-1) = \{\mathbf{p}_i(t|t-1)\}_{i=1}^M$ be the set of M predicted MAV positions by the Kalman trackers at time t. The probabilistic matrix $\mathbf{M}(t) \in \mathbb{R}^{N\times M}$ represents the correlation between $\mathbf{Z}(t)$ and $\mathbf{P}(t|t-1)$. In particular, the ijth element of $\mathbf{M}(t)$, denoted as $\mathrm{m}_{ij}(t)$, represents the correlation between the ith observed position $\mathbf{z}_i(t)$ and the jth predicted position $\mathbf{p}_j(t|t-1)$. It is defined as

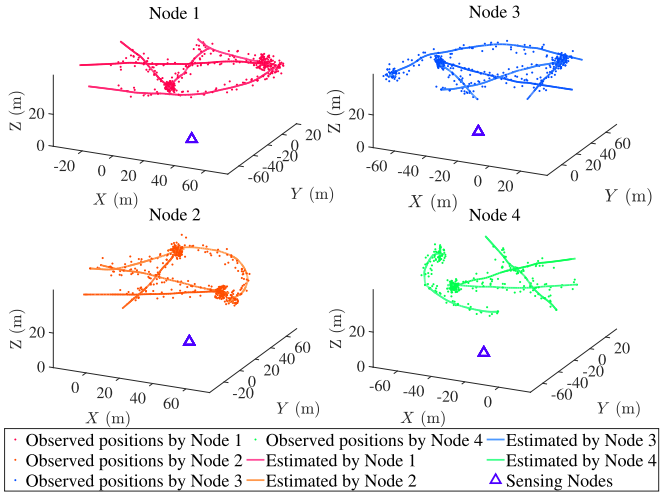
$$\mathrm{m}_{ij}(t) = \begin{cases} 0, & \text{if } d_{ij}(t) > d_{\max} \\ \dfrac{1}{1 + d_{ij}(t)}, & \text{if } d_{ij}(t) \le d_{\max}, \end{cases} \tag{13}$$

where $d_{ij}(t) = \|\mathbf{z}_i(t) - \mathbf{p}_j(t|t-1)\|$ is the Euclidean distance between $\mathbf{z}_i(t)$ and $\mathbf{p}_j(t|t-1)$. In equation (13), $d_{\max}$ is a distance threshold. The larger the $\mathrm{m}_{ij}(t)$, the higher the possibility that $\mathbf{z}_i(t)$ and $\mathbf{p}_j(t|t-1)$ belong to the same

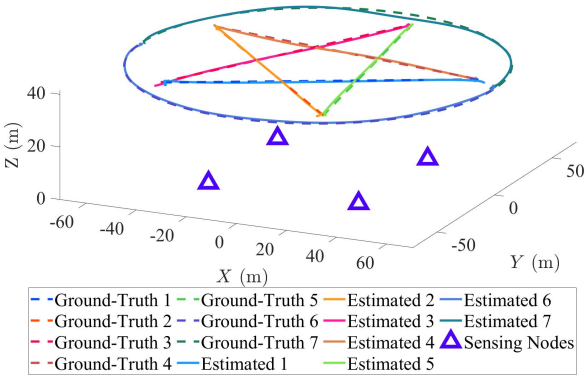


(a) Estimated trajectory (b) Refined trajectory

Fig. 13. Comparison between the estimated trajectories before and after trajectory refinement. In the simulation, the observed data points are sampled in a normal distribution with $\mu = 0, \sigma = 10$.



(a) Estimated trajectories by each sensing node



(b) Fused trajectories of all sensing nodes

Fig. 14. Simulation results of 4 sensing nodes tracking 7 MAVs. Due to the limited observation distance, each sensing node in (a) can only track a part number of MAVs. In (b), by fusing the trajectory segments obtained from the sensing nodes, the global trajectories are constructed.

MAV. The optimal matching result can be obtained by the Kuhn-Munkres algorithm.

More details for state estimation and data association are given below. If a Kalman tracker is matched with an observed position, it will update the MAV trajectory based on the observed position. If a Kalman tracker matches no observations, it simply updates the MAV trajectory by pure prediction without observations. When a Kalman tracker matches no observations for a period of time, it will vanish. If an observed position does not match any Kalman tracker, we consider it as a new MAV and assign it with a new Kalman tracker.

Although the trajectories given by the Kalman trackers are smoother than the original observed MAV positions, they are still not satisfactory. See Fig. 13(a) for illustration. Especially, observation outliers may cause spikes in the trajectories. The spikes in the estimated trajectory segments may cause severe problems in trajectory fusion among multiple sensing nodes as discussed in the next subsection. It is, therefore, necessary to further refine the trajectories.

To do that, we assume that the velocity direction of a MAV does not change vastly within a short period Δt_s. Suppose the current time is t. Then, the estimated position $\mathbf{p}(t')$ for any $t' \in [t - \Delta t_s, t]$ is refined as

$$\mathbf{p}(t') \leftarrow \alpha(t')\mathbf{g}(t') + [1 - \alpha(t')]\mathbf{p}(t'). \tag{14}$$

In equation (14), $\alpha(t') = (t - t')/\Delta t_s$ and $\mathbf{g}(t') = (t' - t + \Delta t_s)[\mathbf{p}(t) - \mathbf{p}(t - \Delta t_s)]/\Delta t_s$.

It is clear that the refined $\mathbf{p}(t')$ is a weighted average of $\mathbf{g}(t')$ and $\mathbf{p}(t')$. Here, $\mathbf{g}(t')$ is a vector pointing from $\mathbf{p}(t - \Delta t_s)$ to $\mathbf{p}(t)$ and hence interpreted as the average velocity direction over $[t - \Delta t_s, t]$. The weight $\alpha(t') \to 1$ when $t' \to t - \Delta t_s$ and $\alpha(t') \to 0$ when $t' \to t$. Therefore, selecting larger values of Δt_s would make the trajectory smoother but may also average out the motion information of the MAV. It is a tradeoff when selecting Δt_s. In our case, we select Δt_s as 6 s according to our experimental scenarios. A simulation example is given in Fig. 13(b) to illustrate the effectiveness of the refinement method.

B. Trajectory Fusion Over Multiple Sensing Nodes

In the previous subsection, the MAV trajectories have been obtained in each sensing node. Since one MAV may be observed by multiple sensing nodes, we next fuse these trajectories to identify the total number of MAVs and their global trajectories.

The first step is to convert the trajectories obtained in each sensing node to a common global coordinate frame. The conversion is based on the attitude and position measurements provided by the IMU and GPS on each sensing node. Since the conversion is trivial, the details are omitted here. In the following, all the trajectories are expressed in the common global coordinate frame.

Suppose $\{\tau_i^k(t)\}$ is the set of the trajectory segments provided by node k at time t, and $\{\tau_j^F(t)\}$ is the set of fused trajectories at t. Note that $\{\tau_i^k(t)\}$ are trajectories over the time interval $[t - \Delta t, t]$. Here, Δt means the length of the trajectories. We select Δt as 6 s in our work considering the communication capability and computational efficiency.

TABLE VII
THE PERFORMANCE OF DIFFERENT MULTI-TARGET TRACKING ALGORITHMS

Algorithm	Node 1 (6 targets)			Node 2 (6 targets)		
	position RMSE (m)	velocity RMSE (m/s)	Runtime (ms)	position RMSE (m)	velocity RMSE (m/s)	Runtime (ms)
CV+GNN	2.0652	4.8892	14.6012	2.5271	5.6899	13.2343
CV+GNN+Refinement (Ours)	1.3946	1.5689	14.9781	1.8837	1.9152	13.5639
IMM+GNN	1.7760	2.8915	87.4625	2.3863	3.6973	75.0003
IMM+GNN+Refinement	1.2313	1.1349	87.8010	1.8971	1.6529	75.3294
Algorithm	Node 3 (5 targets)			Node 4 (6 targets)		
	position RMSE (m)	velocity RMSE (m/s)	Runtime (ms)	position RMSE (m)	velocity RMSE (m/s)	Runtime (ms)
CV+GNN	2.0905	4.2104	11.2876	2.4057	5.3874	13.4468
CV+GNN+Refinement (Ours)	1.7739	1.6799	11.5478	1.6197	1.8134	13.7891
IMM+GNN	2.0037	3.0187	56.6449	2.5232	3.8045	72.8935
IMM+GNN+Refinement	1.6266	1.3951	56.8980	1.9321	1.8062	73.2047

To fuse the trajectory segments to the global trajectories, we adopt the GNN algorithm to associate $\{\tau_i^k(t)\}$ with $\{\tau_j^F(t)\}$. If $\tau_i^k(t)$ is associated with $\tau_j^F(t)$, then $\tau_i^k(t)$ would be used to update $\tau_j^F(t)$. To generate the trajectory of the jth MAV, a metric to measure the similarity between the trajectory segments and the existing global trajectories is needed.

It is noticed that the trajectory of a MAV provided by a sensing node may be biased due to the location or attitude measurement error of that node. As a result, trajectory matching based purely on position errors may fail. To address this problem, we notice that, when two trajectory segments correspond to the same MAV, the position and velocity should be similar at each time step. Thus, we define the similarity score $\mathrm{s}_{ij}^k(t)$ as

$$\mathrm{s}_{ij}^k(t) = \gamma \frac{1}{1+d_v(t)} + (1-\gamma)\frac{1}{1+d_p(t)}, \tag{15}$$

where $\gamma \in (0,1)$ is a trade-off ratio. In equation (15), $d_p(t)$ and $d_v(t)$ are the averaged position and velocity errors of the two trajectories over $[t-\Delta t, t]$. In particular, they are defined as

$$d_p(t') = \frac{1}{\Delta t/\delta t}\sum_{t'=t-\Delta t}^{t} \left\|\mathbf{p}_i^k(t') - \mathbf{p}_j^F(t')\right\|, \tag{16}$$

$$d_v(t') = \frac{1}{\Delta t/\delta t}\sum_{t'=t-\Delta t}^{t} \left\|\mathbf{v}_i^k(t') - \mathbf{v}_j^F(t')\right\|, \tag{17}$$

where δt denotes the sampling time. In equation (16), $\mathbf{p}_i^k(t')$ and $\mathbf{p}_j^F(t')$ are the positions of the points in $\tau_i^k(t)$ and $\tau_j^F(t)$ at time t', respectively. In equation (17), $\mathbf{v}_i^k(t')$ and $\mathbf{v}_j^F(t')$ are the velocities of the points in $\tau_i^k(t)$ and $\tau_j^F(t)$ at time t', respectively.

The higher the similarity score, the higher the probability that $\tau_i^k(t)$ corresponds to $\tau_j^F(t)$. The optimal matching among $\{\tau_i^k(t)\}$ and $\{\tau_j^F(t)\}$ can be obtained by the Kuhn-Munkres algorithm. If $\tau_i^k(t)$ does not associate with any existing trajectories, then we will construct a new global trajectory. If $\tau_j^F(t)$ does not associate with any trajectory segments provided by the sensing nodes for a sufficiently long period of time, then the $\tau_j^F(t)$ will be deleted. The fused global trajectories could be further refined by the trajectory refinement method proposed in the last subsection.

C. Simulation Verification

1) Experimental Setup: To evaluate the performance of the proposed algorithm, we conduct a simulation experiment where there are 7 MAVs and 4 sensing nodes. The MAVs all fly at an altitude of 40 m with a speed of 4 m/s. As shown in Fig. 14, the trajectories of the MAVs form a pentagram in a circle with a radius of 70 m. The locations of the sensing nodes are $(30, 30, 0)$, $(30, -30, 0)$, $(-30, -30, 0)$, and $(-30, 30, 0)$, respectively. The maximum observation range of each sensing node is set to 70 m, beyond which each sensing node is not able to detect any MAVs. The error of observed data obeys a normal distribution with $\mu = 0, \sigma = 10$.

We take the root mean square error (RMSE) as a statistical metric [48], [49] to measure the difference between the estimated global trajectories and the ground-truth trajectories. The averaged position RMSE, averaged velocity RMSE, and runtime are adopted to evaluate the performance of the algorithm.

2) Verification of the Trajectory Refinement Module: To verify the capability of the proposed trajectory refinement module, we adopt four multi-target tracking algorithms for comparison. They are CV+GNN, CV+GNN+Refinement, IMM+GNN, and IMM+GNN+Refinement. Here, CV denotes the Kalman tracker based on the constant velocity model, IMM denotes the interactive multiple model filter, GNN denotes the global nearest neighbor algorithm, and Refinement denotes the trajectory refinement module.

The testing results are given in Table VII. It can be seen from the results of CV+GNN and CV+GNN+Refinement that, after being processed by the trajectory refinement module, the averaged position RMSE and velocity RMSE drop significantly while the runtime increases a little, demonstrating the effectiveness of the proposed trajectory refinement module. By comparing CV+GNN+Refinement with IMM+GNN+Refinement, we find that CV+GNN+Refinement has a similar position RMSE and velocity RMSE with

TABLE VIII
THE PERFORMANCE OF THE PROPOSED SYSTEM IN TRACKING 7 TARGETS

Target	1	2	3	4	5	6	7
position RMSE (m)	1.2344	1.1483	1.9766	1.0925	1.8200	1.2354	1.4286

IMM+GNN+Refinement, but less running time. Thus, we adopt CV+GNN+Refinement in the proposed framework.

3) Verification of the Trajectory-Based MAV Tracking Algorithm: As shown in Fig. 14, with the proposed algorithm, each sensing node could successfully track multiple targets even though the trajectories may be incomplete due to limited sensing range. The fused trajectories by all the sensing nodes are complete and more accurate. It can be seen in Table VIII that the averaged position RMSE of each target is about 1.4 m.

Some remarks about the trajectory-based MAV tracking algorithm are given below. First, the standard data association algorithm is based on an assumption that each target generates at most one measurement, while in our case the target MAV can be detected by multiple sensing nodes at the same time. Thus, we fuse the trajectory segments obtained by each sensing node with all global trajectories respectively. Second, since the modules of the trajectory-based MAV tracking algorithm are independent, they can be replaced by other advanced algorithms in the future.

VII. EXPERIMENTAL VERIFICATION OF THE OVERALL SYSTEM

To verify the effectiveness of the overall system, two experiments with different numbers of sensing nodes and MAVs are conducted. The first is one sensing node detecting one MAV. The second is multiple sensing nodes detecting multiple MAVs. The experiment setup is shown in Fig. 1.

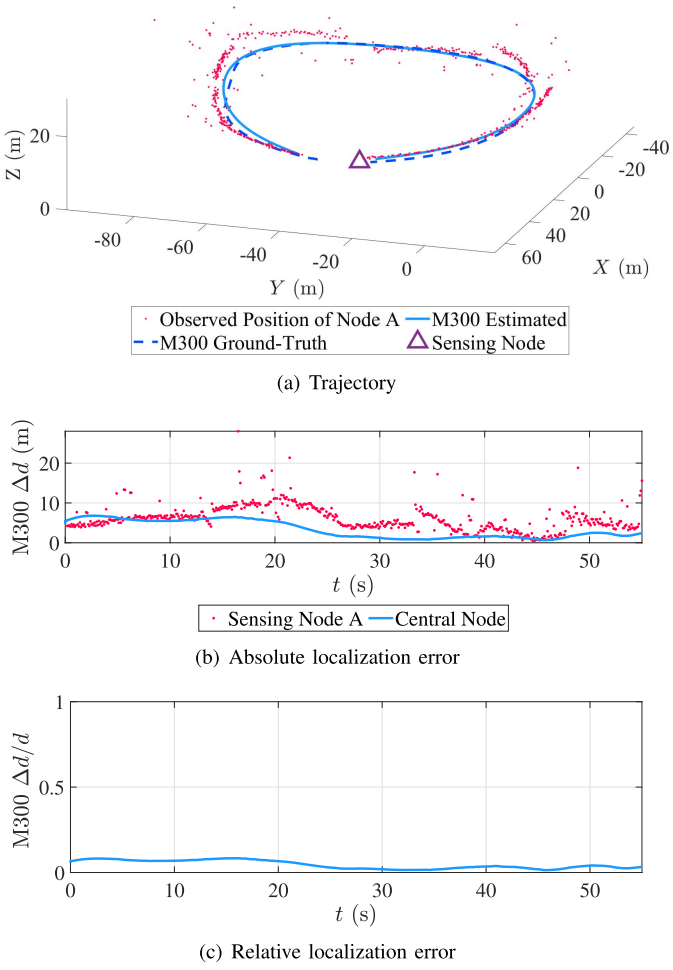
The objects detected and localized by each sensing node are represented as 3D points, which will either be fused into the existing global trajectories or used to create new global trajectories. We thus evaluate the capability of the system by comparing these global trajectories with the ground-truth trajectories.

A. Evaluation Metrics

To measure the capability of the entire system, we adopt the absolute localization error, the relative localization error, and the runtime as the metrics.

The absolute localization error $\Delta d(t)$ is defined as $\Delta d(t) = ||p(t) - \hat{p}(t)||$. Here, $p(t)$ denotes the ground-truth position of the MAV, $\hat{p}(t)$ denotes the estimated position of the MAV, and $||\cdot||$ denotes the Euclidean norm of a vector.

The relative localization error is defined as $\Delta d(t)/d(t)$. Here, $\Delta d(t)$ denotes the absolute localization error. $d(t)$ denotes the averaged ground-truth distance between the target MAV and the sensing nodes. It is defined as $d(t) = 1/N \sum_{i=1}^{N} ||p(t) - p_i||$. Here, N denotes the total number of the sensing nodes, $p(t)$ denotes the ground-truth position of the MAV, and p_i denotes the ground-truth position of the sensing node i. The ground-truth position of the sensing node and the target MAV is obtained by the RTK-GPS equipped on them.



(a) Trajectory

(b) Absolute localization error

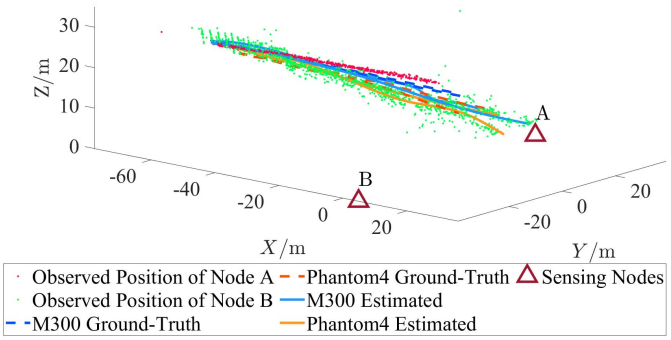
(c) Relative localization error

Fig. 15. Experimental results of one sensing node detecting one MAV. In (a), the estimated trajectory is compared with the ground-truth trajectory. In (b), the scatters are the localization error of the stereo vision. The lines in (b) and (c) are the errors between the ground-truth trajectory and the estimated trajectory.

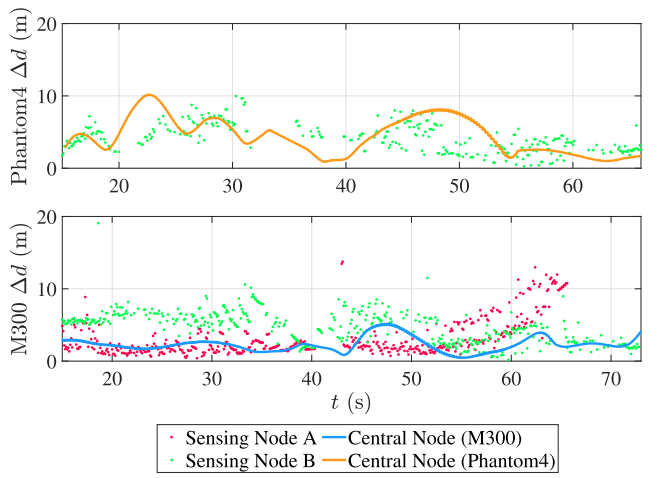
B. Evaluation Results

1) One Sensing Node Detecting One MAV: In the first experiment, an M300 MAV (see Fig. 6) flies along a circular trajectory around one sensing node. The distance from M300 to the sensing node is from 40 to 65 m.

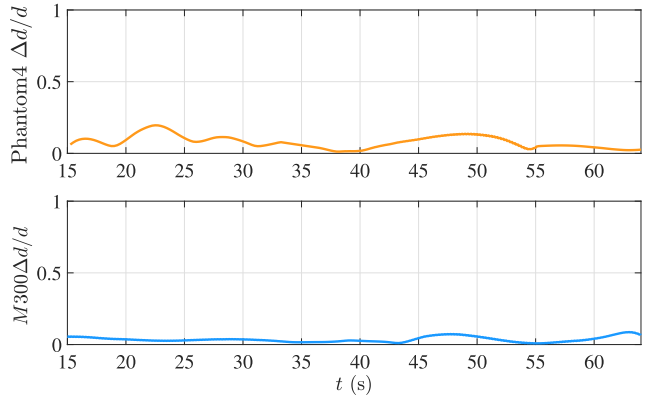
The experimental results are shown in Fig. 15. As can be seen, the MAV is successfully detected, localized, and tracked by the proposed system. The absolute and relative localization errors are shown in Fig. 15(b) and Fig. 15(c), respectively. As can be seen, the average absolute localization



(a) Trajectory



(b) Absolute localization error



(c) Relative localization error

Fig. 16. Experimental results of two sensing nodes detecting two MAVs. In (a), the trajectory of M300 and Phantom 4 is estimated by the proposed system and compared with their ground-truth trajectory. In (b), the scatters are the localization error of the stereo vision. The lines in (b) and (c) are the errors between the ground-truth trajectory and the estimated trajectory.

error Δd is 3.34 m and the average relative localization error is 4.57%.

2) Two Sensing Nodes Detecting Two MAVs: In this experiment, we use M300 and Phantom 4 as target MAVs (see Fig. 6). They fly in the same straight line one after another. This is a challenging scenario because the trajectories of the two MAVs may overlap spatially. They can only be distinguished temporally, requiring high performance of the system.

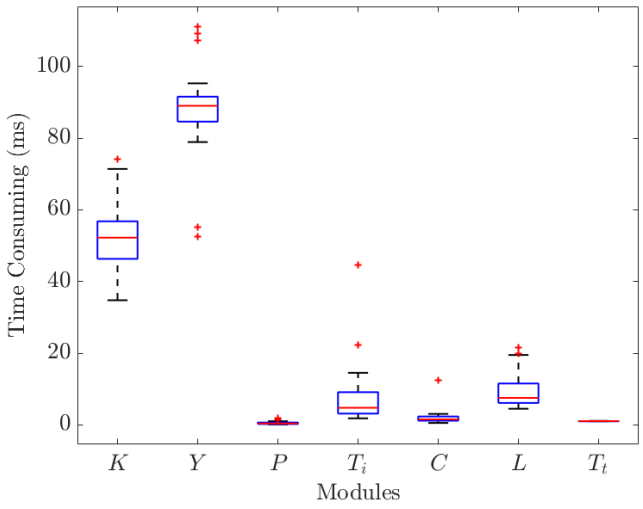


Fig. 17. The running time for each module in the proposed system. The K, Y, P, T_i, C, L, and T_t denote KNN, YOLOv5, pre-screening, multi-target image tracking, classification, localization, and trajectory-based multi-target tracking, respectively.

The experimental results are shown in Fig. 16. It is noticed that M300 could be detected simultaneously by both sensing nodes A and B, but Phantom 4 is detected by node B only but not node A, since it is too far away from node A. As shown in Fig. 16(a), the proposed algorithms could successfully detect, localize, and track the two MAVs. As shown in Fig. 16(b), the average absolute localization errors of M300 and Phantom 4 are 2.39 m and 4.52 m, respectively. The average relative localization errors, as shown in Fig. 16(c), are 4.05% and 7.81%, respectively.

3) Computational Efficiency: To verify the computation efficiency of the overall system, we evaluate the consuming time of each module in an experiment and randomly select 30 samples for analysis. In a sensing node, each computer has an Intel i7-11700K @ 3.6 GHz CPU and Nvidia RTX 3070 GPU. Except for YOLOv5 implemented on the GPU, other modules run on the CPU. The resolution of the stitched image processed by each computer is 5760 × 1280. In the central node, the computer has an Intel i7-10750H @ 2.6 GHz CPU.

The evaluation result is shown in Fig. 17. As can be seen, the MAV detection module costs the most time in the proposed system. It cost around 97.82 ms. While the MAV localization and trajectory-based tracking module cost around 9.99 ms and 1.03 ms, respectively. Since the running time of the MAV detection module does not depend on the number of MAVs and the other modules cost little time, the running speed of the proposed system could reach at least 9 FPS.

Some remarks about the effective range of the proposed system are given below. The effective range of the entire system is jointly determined by the three modules: MAV detection, MAV localization, and trajectory-based MAV tracking. Regarding the detection algorithm, its effective detection range of M300 can reach 150 m. However, when we incorporate the other two algorithms, the overall effective range decreases to 80 m.

VIII. Conclusion

This paper presented a novel system of panoramic stereo camera networks system for MAV detection, localization, and tracking. Each sensing node in the system is a 360° panoramic stereo camera. The effective sensing range of the entire system for a MAV like DJI M300 could reach 80 m, much higher than conventional stereo cameras. With multiple sensing nodes, the system could secure a large area against the intrusion of malicious MAVs. Extensive experiments verified the effectiveness and efficiency of the proposed system.

Acknowledgment

The authors would like to thank their group mates, Yongqi Li, Yize Mi, Zhikun Wang, Kang Li, Zian Ning, and Yin Zhang, for their great support especially their help in the flight experiments.

References

[1] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi, and J. Chen, "Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 68–74, Apr. 2018.

[2] Y. Tang *et al.*, "Vision-aided multi-UAV autonomous flocking in GPS-denied environment," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 616–626, Jan. 2019.

[3] 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.*, vol. 50, no. 6, pp. 2590–2603, Jun. 2020.

[4] G. Skorobogatov, C. Barrado, and E. Salamí, "Multiple UAV systems: A survey," *Unmanned Syst.*, vol. 8, no. 2, pp. 149–169, 2020.

[5] J. Ren and X. Jiang, "A three-step classification framework to handle complex data distribution for radar UAV detection," *Pattern Recognit.*, vol. 111, Mar. 2021, Art. no. 107709.

[6] J. Xie, C. Gao, J. Wu, Z. Shi, and J. Chen, "Small low-contrast target detection: Data-driven spatiotemporal feature fusion and implementation," *IEEE Trans. Cybern.*, early access, May 24, 2021, doi: 10.1109/TCYB.2021.3072311.

[7] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao, "Air-to-air visual detection of micro-UAVs: An experimental evaluation of deep learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1020–1027, Apr. 2021.

[8] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, 2006.

[9] G. Jocher *et al.*, "Ultralytics/YOLOV5: V5.0—YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations," Apr. 2021.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2564–2571.

[11] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[12] F. Lin, K. Peng, X. Dong, S. Zhao, and B. M. Chen, "Vision-based formation for UAVs," in *Proc. 11th IEEE Int. Conf. Control Automat. (ICCA)*, Jun. 2014, pp. 1375–1380.

[13] F. Gökçe, G. Üçoluk, E. Şahin, and S. Kalkan, "Vision-based detection and distance estimation of micro unmanned aerial vehicles," *Sensors*, vol. 15, no. 9, pp. 23805–23846, 2015.

[14] K. R. Sapkota *et al.*, "Vision-based unmanned aerial vehicle detection and tracking for sense and avoid systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1556–1561.

[15] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman, "Multi-target detection and tracking from a single camera in unmanned aerial vehicles (UAVs)," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4992–4997.

[16] S. Minaeian, J. Liu, and Y.-J. Son, "Effective and efficient detection of moving targets from a UAV's camera," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 497–506, Feb. 2018.

[17] Y. Cao, Z. Zhang, Y. Fan, M. Ding, and J. Tao, "Vision-based flying targets detection via spatiotemporal context fusion," *IEEE Access*, vol. 7, pp. 144090–144100, 2019.

[18] Y. Shu, Y. Sui, S. Zhao, Z. Cheng, and W. Liu, "Small moving object detection and tracking based on event signals," in *Proc. 7th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2021, pp. 792–796.

[19] M. P. Muresan, S. Nedevschi, and R. Danescu, "Robust data association using fusion of data-driven and engineered features for real-time pedestrian tracking in thermal images," *Sensors*, vol. 21, no. 23, p. 8005, Nov. 2021.

[20] E. Unlu, E. Zenou, N. Riviere, and P.-E. Dupouy, "Deep learning-based strategies for the detection and tracking of drones using several cameras," *IPSJ Trans. Comput. Vis. Appl.*, vol. 11, no. 1, pp. 1–13, Dec. 2019.

[21] M. W. Ashraf, W. Sultani, and M. Shah, "Dogfight: Detecting drones from drones videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7067–7076.

[22] R. Opromolla and G. Fasano, "Visual-based obstacle detection and tracking, and conflict detection for small UAS sense and avoid," *Aerosp. Sci. Technol.*, vol. 119, Dec. 2021, Art. no. 107167.

[23] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, "Monocular depth estimation based on deep learning: An overview," *Sci. China Technol. Sci.*, vol. 63, no. 9, pp. 1–16, 2020.

[24] M. Gavish and A. J. Weiss, "Performance analysis of bearing-only target location algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 3, pp. 817–828, Jul. 1992.

[25] Z. Berman, "A reliable maximum likelihood algorithm for bearing-only target motion analysis," in *Proc. IEEE Conf. Decis. Control (CDC)*, vol. 5, Dec. 1997, pp. 5012–5017.

[26] Z. Wang, J.-A. Luo, and X.-P. Zhang, "A novel location-penalized maximum likelihood estimator for bearing-only target localization," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6166–6181, Dec. 2012.

[27] S. Hinterstoisser *et al.*, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2012, pp. 548–562.

[28] K. Wada, E. Sucar, S. James, D. Lenton, and A. J. Davison, "MoreFusion: Multi-object reasoning for 6D pose estimation from volumetric fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, p. 14.

[29] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[30] J. Ding, Z. Yan, and X. We, "High-accuracy recognition and localization of moving targets in an indoor environment using binocular stereo vision," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 4, p. 234, Apr. 2021.

[31] J. Xin, X. Ma, Y. Deng, D. Liu, and H. Liu, "A new method of stereo localization using dual-PTZ-cameras," in *Proc. Int. Conf. Intell. Robot. Appl. (ICIRA)*, 2012, pp. 460–472.

[32] I. Ullah, T. L. Song, and T. Kirubarajan, "Active vehicle protection using angle and time-to-go information from high-resolution infrared sensors," *Opt. Eng.*, vol. 54, no. 5, May 2015, Art. no. 053110.

[33] J. Yi and S. Srigrarom, "Near-parallel binocular-like camera pair for multi-drone detection and 3D localization," in *Proc. 16th Int. Conf. Control, Automat., Robot. Vis. (ICARCV)*, Dec. 2020, pp. 204–210.

[34] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.

[35] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Syst.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.

[36] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. AC-24, no. 6, pp. 843–854, Dec. 1979.

[37] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3047–3055.

[38] S. Taguchi and K. Kidono, "Exclusive association sampling to improve Bayesian multi-target tracking," *IEEE Access*, vol. 8, pp. 193116–193127, 2020.

[39] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4696–4704.

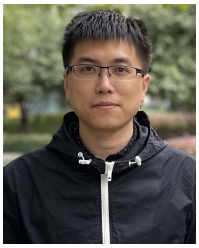
[40] A. T. Kamal, J. H. Bappy, J. A. Farrell, and A. K. Roy-Chowdhury, "Distributed multi-target tracking and data association in vision networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1397–1410, Jul. 2015.

[41] N. S. J. Liang and S. Srigrarom, "Multi-camera multi-target drone tracking systems with trajectory-based target matching and re-identification," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2021, pp. 1337–1344.

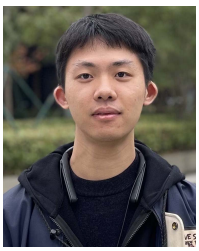
[42] *OpenCV*. Accessed: Jun. 2021. [Online]. Available: https://opencv.org

[43] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[44] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020, vol. 34, no. 7, p. 12993–13000.
[45] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
[46] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7363–7372.
[47] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[48] T. L. Song and D. Mušicki, "Smoothing innovations and data association with IPDA," *Automatica*, vol. 48, no. 7, pp. 1324–1329, Jul. 2012.
[49] D. Musicki and S. Suvorova, "Tracking in clutter using IMM-IPDA-based algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 111–126, Jan. 2008.

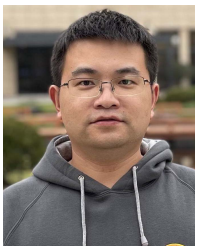


Ye Zheng received the B.S. degree from the Guilin University of Technology, Guilin, China, in 2016, and the M.S. degree from the Department of Electromechanical and Automation, Harbin Institute of Technology, Shenzhen, China, in 2019. He is currently pursing the Ph.D. degree in computer science with the School of Engineering, Westlake University, Hangzhou, China, and joint-cultivated by Zhejiang University, Hangzhou. His research interests lie in computer vision and robotics.



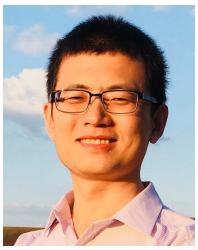
Canlun Zheng received the B.S. degree in mechanical engineering from Shandong University, Shandong, China, in 2017, and the M.S. degree from the Department of Astronautics Engineering, Beihang University, Beijing, China, in 2020. He is currently pursing the Ph.D. degree in computer science with the School of Engineering, Westlake University, Hangzhou, China, and joint-cultivated by Zhejiang University, Hangzhou. His research interests include UAV control, multitarget tracking, and robotics.

Xiaoyu Zhang received the B.E. and M.E. degrees in mechanical engineering from Beihang University, Beijing, China, in 2017 and 2020, respectively. He is currently pursing the Ph.D. degree in mechanical and automation engineering with The Chinese University of Hong Kong, Hong Kong, China. From 2020 to 2021, he was working as a Research Assistant at the Intelligent Unmanned Systems Laboratory, Westlake University, Hangzhou, China. His research interests include robotic perception, SLAM, and self-driving cars.



Fei Chen received the B.E. degree in mechanical engineering from Oakland University, MI, USA, in 2018. Since 2019, he has been working as a Research Assistant with the Intelligent Unmanned Systems Laboratory, Westlake University. His main job is to assist Ph.D. students in outdoor flight experiments.

Zhang Chen received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2015. He was a Post-Doctoral Fellow with the Graduate School at Shenzhen, Tsinghua University, Shenzhen, China, from 2016 to 2018. He is currently a Research Assistant Professor with the Department of Automation, Tsinghua University, Beijing, China. His research interests include robotics, control systems, and autonomous systems.



Shiyu Zhao (Member, IEEE) received the B.E. and M.E. degrees from the Beijing University of Aeronautics and Astronautics, China, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from the National University of Singapore in 2014. He was a Post-Doctoral Researcher with the Technion—Israel Institute of Technology, Haifa, Israel, and the University of California at Riverside, Riverside, CA, USA, from 2014 to 2016. He was a Lecturer with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K., from 2016 to 2018. He is currently an Associate Professor with the School of Engineering, Westlake University, Hangzhou, China. His research interest lies in theories and applications of aerial robotic systems. He was a co-recipient of the Best Paper Award (Guan Zhao-Zhi Award) in the 33rd Chinese Control Conference, Nanjing, China.