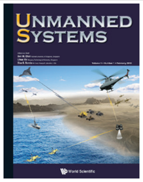
Unnamed Systems (2024)

DOI: 10.1142/S2301385025500360

UNMANNED SYSTEMS
https://www.worldscientific.com/worldscinet/us



A Real-to-Sim-to-Real Approach for Vision-Based Autonomous MAV-Catching-MAV

Zian Ning [*,†], Yin Zhang [†], Xiaofeng Lin [‡], Shiyu Zhao [†,§]

[*] Department of Computer Science & Technology,
Zhejiang University,
Hangzhou 310024, P. R. China

[†] School of Engineering,
Westlake University,
Hangzhou 310024, P. R. China

[‡] Division of Systems Engineering,
Boston University,
Boston, MA 02215, USA

This paper studies the task of vision-based MAV-catching-MAV, where a catcher MAV (micro aerial vehicle) can detect, localize, and pursue a target MAV autonomously. Since it is challenging to develop detectors that can effectively detect unseen MAVs in complex environments, the main novelty of this paper is to propose a *real-to-sim-to-real* approach to address this challenge. In this method, images of real-world environments are first collected. Then, these images are used to construct a high-fidelity simulation environment, based on which a deep-learning detector is trained. The merit of this approach is that it allows efficient automatic collection of large-scale and high-quality labeled datasets. More importantly, since the simulation environment is constructed from real-world images, this approach can effectively bridge the sim-to-real gap, enabling efficient deployment in real environments. Another contribution of this paper lies in the successful implementation of a fully autonomous vision-based MAV-catching-MAV system including proposed estimation and pursuit control algorithms. While the previous works mainly focused on certain aspects of this system, we developed a completely autonomous system that integrates detection, estimation, and control algorithms on real-world robotic platforms.

Keywords: Real-to-sim-to-real; MAV-catching-MAV.

1. Introduction

This paper studies the task of vision-based MAV-catching-MAV, where a catcher MAV (micro aerial vehicle) can use onboard devices to detect, localize, and pursue a target MAV, and finally catch it by launching a net. This system is interesting to study since it is inspired by the bird-catching-bird behavior in nature [1]. It is also useful since it provides a potential solution for countering misused MAVs [2, 3].

Realizing such a system is highly challenging. First, it requires real-time onboard target detection. Although visual sensing is a promising approach for detecting MAVs, it is still challenging to detect unseen MAVs in complex environments. Second, this system is highly complex since it involves many modules such as detection, estimation, and control. Most existing studies only focus on certain aspects of the system [2–4]. It remains a challenge to realize a complete autonomous MAV-catching-MAV system on real-world robotic platforms.

The main novelty of this paper is to address the two challenges mentioned above. The details are given as follows.

Visual detection of MAVs faces some unique challenges. First, since the target MAV is *non-cooperative*, we do

Received 5 February 2024; Revised 18 March 2024; Accepted 19 March 2024; Published xx xx xx. This paper was recommended for publication in its revised form by editorial board member, Feng Lin.

Email Address: [§] *zhaoshiyu@westlake.edu.cn*

[§] Corresponding author.

not have the datasets of the target MAV in advance. This is more challenging than the cooperative scenario, in which we can train detectors based on the datasets of the cooperative MAVs [2, 5]. Second, since the catcher MAV may observe the target from various angles, the background of the target MAV is dynamically changing and complex (e.g. urban or rural environments). This is more challenging to handle than the static or simple background scenarios [6]. Many existing MAV datasets consist of limited MAV models or background complexity [5, 7], leading to limited generalization capabilities.

Sim-to-real is an effective approach for collecting large-scale datasets and training deep-learning detectors. This approach has three advantages: (1) It can easily incorporate various MAV models and complex environmental models, enhancing dataset diversity; (2) It can automatically and accurately annotate the images, saving plenty of manual labelling efforts; (3) It can conveniently set various conditions, such as observation angles and lighting conditions, further enhancing dataset diversity [8].

However, the sim-to-real approach has a critical limitation that must be overcome: the sim-to-real gap. In particular, simulation environments are different from the real-world environment in many aspects [9]. Detectors trained in a simulated environment may experience a performance downgrade when deployed in the real world. Additional methods such as domain generalization [8], domain adaptation, and transfer learning [10, 11] may be used to alleviate the gap to a certain extent.

Motivated by the sim-to-real gap, this paper proposes a novel *real-to-sim-to-real* approach for MAV detection. In particular, we first collect images of the real-world environment and then construct a high-fidelity simulation environment in the Unreal game engine using these images. Numerous types of MAV models can be efficiently integrated into the simulation environment. Based on the AirSim interface, high-quality datasets can be collected efficiently. For example, we can collect 20,000 images that are accurately labeled and cover various viewing conditions within merely 24 h using one desktop computer.

This real-to-sim-to-real approach inherits the advantages of the sim-to-real approach, such as efficient collection of large amounts of high-quality data. More importantly, since the simulation environment is constructed from real-world images, this approach can alleviate the sim-to-real gap and enhance transfer efficiency. Our experimental results show that the detector trained based on this approach can significantly outperform the detectors trained based purely on simulation or real-world datasets.

An important assumption behind the real-to-sim-to-real approach is that an MAV-catching-MAV system is deployed in a *specific* region, even though the region may cover various scenes. This assumption is often valid in practice. We can fully utilize the prior knowledge of the specific region, rather than aiming to develop a one-size-fits-all system, which is not only difficult to achieve but also unnecessary. For example, using the environment model in the simulation can extend the range of view angles of perceiving the environment. Real images only contain limited view angles, whereas in simulation, any view angle is possible. With this idea, the proposed approach enhances specificity, leading to better sim-to-real transfer efficiency.

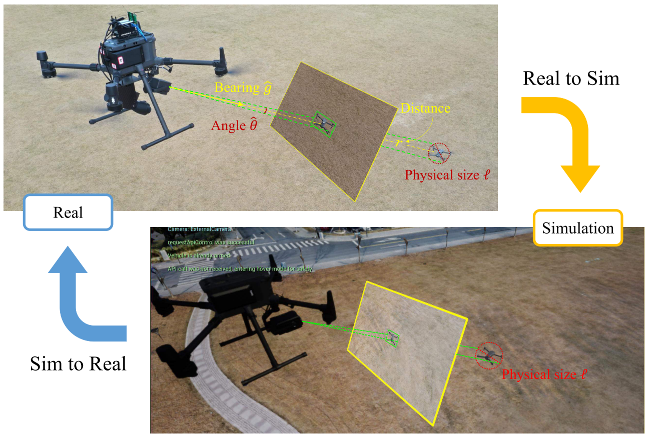


Fig. 1. An illustration of the *Real-to-Sim-to-Real* approach. The simulation environment is constructed using real-world images. The real-world implementation utilizes the detector trained in the simulation.

To the best of our knowledge, our work is the first to propose the real-to-sim-to-real approach for MAV detection. There are some relevant but different works in the literature about data augmentation. For example, MAV models can be rendered on real-world images to generate realistic augmented data [12–14]. Different from these approaches, we use real-world images to construct a high-fidelity simulation environment and then collect data. One merit of doing this is that the constructed 3D environment can be used to automatically acquire MAV images with various viewing angles and distances, leading to high flexibility of data generation. This real-to-sim-to-real approach may also be applicable to similar challenging tasks other than MAV detection tasks as long as the vision task is performed in a specific region.

Realization of the entire system: The vision-based MAV-catching-MAV system is complex since it involves a series of interconnected algorithms such as target detection, gimbal tracking, motion estimation, and pursuit control. It is also nontrivial to develop a robotic system to integrate the algorithms all together. Up to now, the existing studies [2, 15, 16] including our previous work [4] merely focus on certain aspects of such systems.

For the unknown target's motion estimation problem, bearing-only based estimator [4] and bearing-angle based estimator [17] are two existing methods. However, both

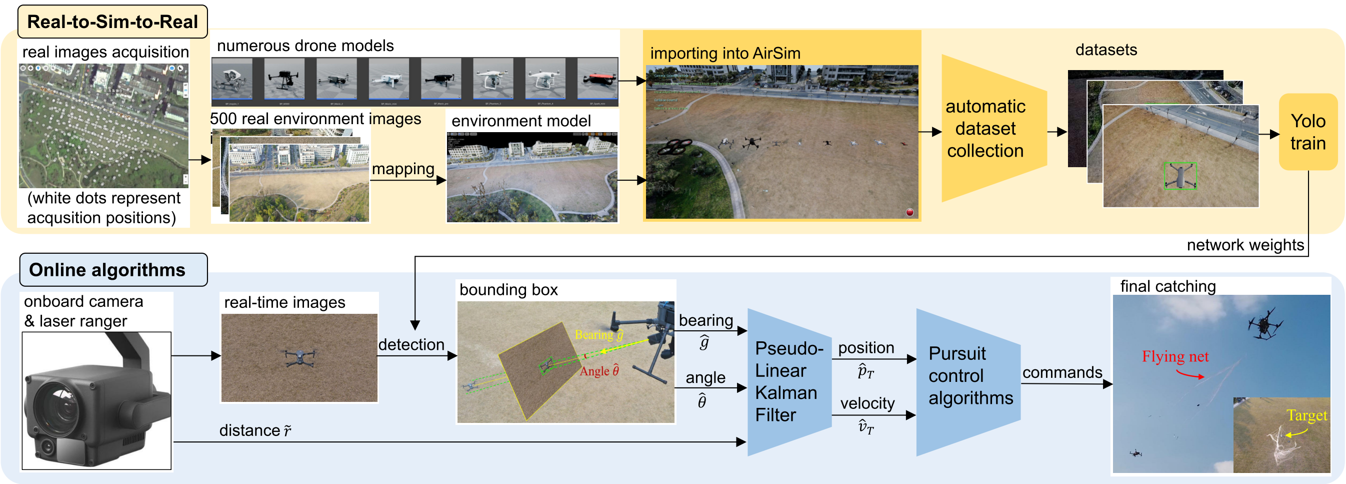


Fig. 2. The structure of the proposed autonomous MAV-catching-MAV system.

methods encounter observability problems. We integrate vision and laser ranger for target motion estimation. The proposed estimator can estimate the motion of unknown-sized targets. A pursuit control algorithm is proposed based on the Lyapunov energy theory [18]. The proposed algorithm enables the catcher MAV to track the target stably at the desired distance.

The second contribution of this work is to realize a complete vision-based MAV-catching-MAV system that incorporates all necessary modules. The catcher MAV can automatically detect, pursue, and finally catch a target MAV. All proposed algorithms can be executed onboard in real-time. Both simulation and real-world experiments validate the effectiveness of the system.

2. System Overview

The architecture of the MAV-catching-MAV system is shown in Fig. 2. It contains two parts. The first part is the real-to-sim-to-real approach for MAV detection. Its purpose is to generate a high-performance MAV detector. This part is introduced in detail in Sec. 3.

The second part is to design and integrate all the algorithms on real robotic platforms. In particular, the catcher MAV is a DJI M300 quadcopter which is a reliable mature commercial product with the ability for accurate RTK self-positioning. It also supports DJI APIs. These features make it well-suited for the MAV-catching-MAV system. The onboard gimballed H20 camera integrates a single-beam laser ranger which can provide distance measurements when the target appears at the center of the image. A pseudo-linear target motion estimation algorithm is proposed to fuse the measurements. The estimation and control algorithms are introduced in Sec. 4.

To verify the effectiveness of the system, we construct a simulation system based on the Unreal game engine and AirSim. All algorithms including detection, estimation, and control are tested (Sec. 5). Outdoor experiments also validate the effectiveness of the system (Sec. 6).

3. Real-to-Sim-to-Real Visual Detection

This section introduces the real-to-sim-to-real approach for non-cooperative MAV detection in a specified environment.

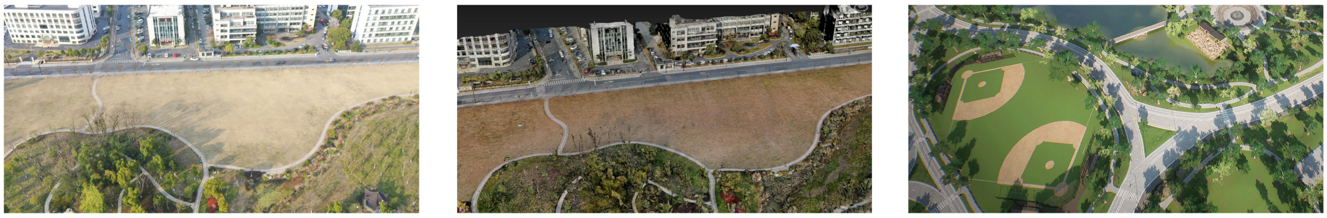
3.1. *Overview approach*

The framework of the real-to-sim-to-real approach is shown in the upper block of Fig. 2. There are three steps in the approach.

(1) *Real-to-sim*: First, we need to collect sufficient images of the specified environment where the MAV detection task happens. Then, these images are used to construct a high-fidelity model of the environment. Finally, the environment model and numerous real MAV models are imported into the AirSim platform.

In our work, the specified environment is around a park named Yunqi. The dimensions of the environment that we aim to cover are 360 × 280 × 40 m. This park has a big area of lawn, which is surrounded by some buildings and small mountains (Fig. 3(a)). Such an environment, which involves both urban and plant scenes, is of high scene complexity.

We collected 500 images of the environment. The locations where these images are captured by the camera can roughly cover the entire park (see the up-left corner of Fig. 2). The images are used to construct a high-fidelity



(a) Real-world environment. (b) Constructed real-to-sim environment. (c) A pure simulation environment.

Fig. 3. The three types of environments.

digital model by the DJI Terra software (Fig. 3(b)). Then, the model is imported into the AirSim platform.

We also incorporate eight MAV models into the AirSim platform (Fig. 5). These models range in size from 0.17 m (DJI Spark) to 0.895 m (DJI M300). The models can be downloaded from the CGTrader website, where plenty of realistic MAV models are available.

(2) *Automatic dataset collection:* In the AirSim platform, we can conveniently set up various conditions such as the target's position and attitude, viewing angle and distance, and lighting conditions. A wide range of these conditions can improve the diversity of the collected datasets. Moreover, since the ground truth of the bounding box can be obtained in AirSim, the dataset can be labelled efficiently and accurately. For example, 20,000 images that are well-labeled and cover a wide range of conditions can be collected automatically within 24 hours.

(3) *Detector training:* A Yolo-based detector is trained using the datasets collected in the simulation environment. The trained detector can then be deployed in the real world without any changes to achieve vision-based MAV-catching-MAV. The details of the datasets are given in the following.

3.2. *Datasets*

Six datasets are collected: three training datasets and three test datasets. The details are listed in Table 1. Some explanation about the datasets is given as follows.

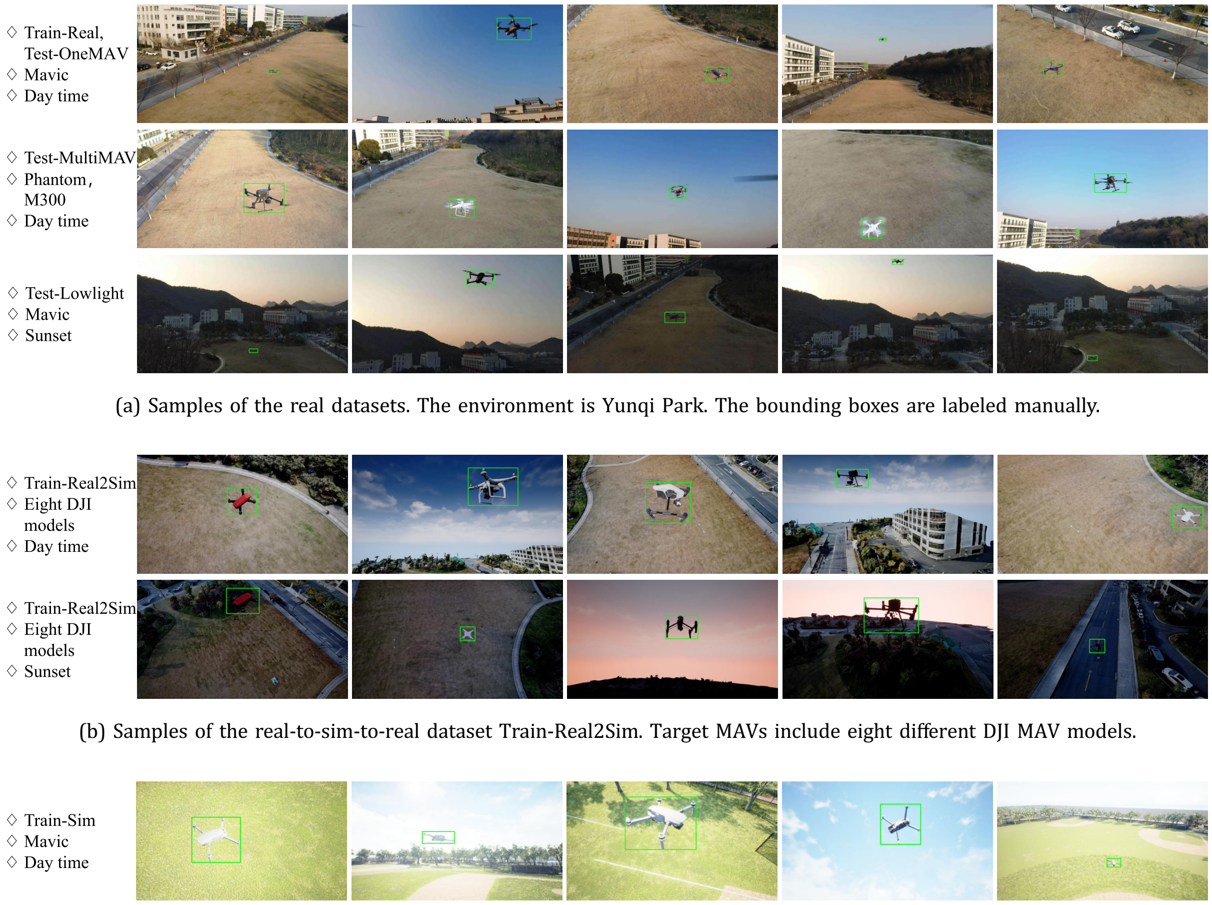
Dataset Train-Real is collected in the real-world environment during daytime. This dataset involves merely one target MAV (DJI Mavic2). Samples are given in the first row of Fig. 4(a). **Dataset Train-Real2Sim** is collected in the simulation environment constructed from real-world images. Eight MAV models (Fig. 5) are used as the target MAVs. Different viewing angles and light conditions are considered when the dataset is collected. Samples of the dataset are shown in Fig. 4(b). **Dataset Train-Sim** is collected in a conventional simulation environment that is NOT constructed from real-world images. In particular, we use a simulation environment called City Park (Fig. 3(c)) downloaded from the Unreal market. The reason that we select this simulation environment is because it is similar to our real-world park environment. Samples are given in Fig. 4(c). The dataset collection settings for Train-Sim are different from Train-Real2Sim but the same as Test-OneMAV. This setup for Train-Sim is sufficient to illustrate the limitations of the sim-to-real approach (see Table 2).

The above three datasets are used to train three detectors. Then, these detectors are tested on the following testing datasets. All these testing datasets only contain *real-world* images.

Dataset Test-OneMAV is obtained by randomly selecting some samples from Train-Real. As a result, Test-OneMAV shares a similar data distribution to Train-Real. **Dataset Test-MultiMAV** involves two MAVs (Phantom4 and M300). It can be used to test the generalization ability of the detectors for detecting different MAV models. Samples of Test-MultiMAV are given in the second row of Fig. 4(a). **Dataset Test-Lowlight** involves different lighting conditions. It is collected during sunset. As a result, the color of the sky turns orange, and the images are relatively dark. This dataset can be used to test the generalization ability of

Table 1. Key specifications of 6 collected datasets.

Type	Dataset index	Collect approach	Environment	Included MAV model(s)	Light condition	Image count
Training	Train-Real	Real	Yunqi Park	Mavic2	day	1,271
	Train-Real2Sim	Real-to-Sim-to-Real	Yunqi Park model	Eight DJI models	day & sunset	60,000
	Train-Sim	Sim-to-Real	Virtual City Park	Mavic2	day	30,000
Testing	Test-OneMAV	Real	Yunqi Park	Mavic2	day	1,000
	Test-MultiMAV	Real	Yunqi Park	Phantom4 & M300	day	1,000
	Test-Lowlight	Real	Yunqi Park	Mavic2	sunset	1,000



(a) Samples of the real datasets. The environment is Yunqi Park. The bounding boxes are labeled manually.

(b) Samples of the real-to-sim-to-real dataset Train-Real2Sim. Target MAVs include eight different DJI MAV models.

(c) Samples of the sim-to-real dataset Train-Sim. The environment is called CityPark, available in the Unreal Engine marketplace.

Fig. 4. Samples of the collected datasets. The leftmost column gives the specifications of the datasets.

the detectors for different lighting conditions. Samples of Test-Lowlight are given in the last row of Fig. 4(a).

3.3. *Effectiveness and generalization ability*

We conducted three sets of performance evaluation tests. The mAP results are given in Table 2. The details are explained as follows.

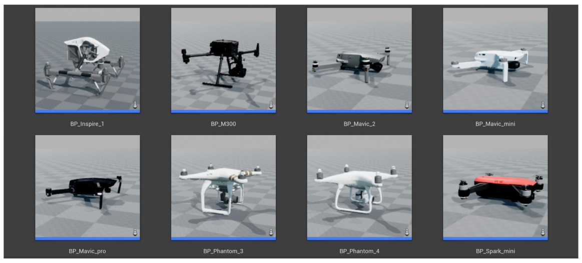


Fig. 5. Eight DJI MAV models were utilized in the collection of Dataset Train-Real2Sim.

The first set of tests aims to verify the effectiveness of the real-to-sim-to-real approach. Please see the second column of Table 2. (1) For the detector trained on Train-Real, the mAP on Test-OneMAV is as high as 99.88%. This is not surprising because Train-Real and Test-OneMAV share similar data distribution. (2) For the detector trained on Train-Sim, the mAP on Test-OneMAV is only 76.35%, which suggests a big sim-to-real gap. (3) For the detector trained on Train-Real2Sim, the mAP on Test-OneMAV is 97.12%. Although it is not as high as the detector trained on Train-Real, it is still satisfactory for practical application and much better than the detector trained on Train-Sim.

Table 2. Test results (mAP@0.5, in %).

Detector trained on	Test-OneMAV	Test-MultiMAV	Test-Lowlight
Train-Real	**99.88**%	92.10%	89.32%
Train-Real2Sim	97.12%	**97.67**%	**96.43**%
Train-Sim	76.35%	55.99%	37.18%

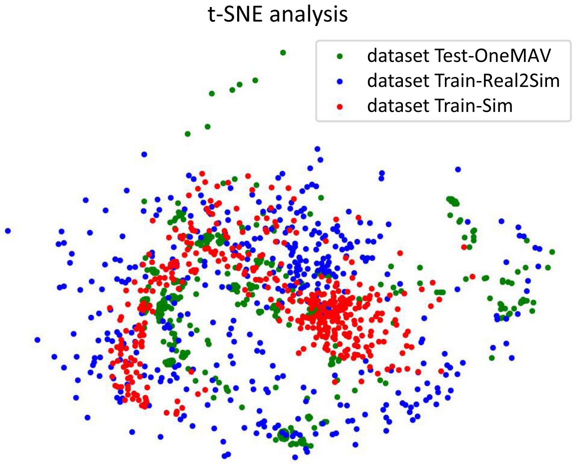


Fig. 6. The results of t-SNE domain analysis.

It verifies that the real-to-sim-to-real approach can effectively alleviate the sim-to-real gap. More importantly, this detector has stronger generalization abilities as analyzed below.

The second set of tests aims to study the generalization ability for *different MAV models*. Please see the third column of Table 2. For the detector trained on Train-Real2Sim, the mAP on Test-MultiMAV is 97.67%. It is more than 5% higher than the detector trained on Train-Real. By contrast, for the detector trained on Train-Sim, the mAP on Test-MultiMAV is as low as 55.99%.

The third set of tests aims to demonstrate the generalization ability for *different lighting conditions*. As shown in the last column of Table 2, for the detector trained on Train-Real2Sim, the mAP on Test-Lowlight is 96.43%, which is about 7% higher than the detector trained on Train-Real. For the detector trained on Train-Sim, the mAP on Test-Lowlight is merely 37.18% due to the sim-to-real gap. Of course, the sim-to-real gap can be leveraged to a certain extent using domain adaptation techniques. Our work shows that the simple idea of real-to-sim-to-real can effectively alleviate the sim-to-real gap.

3.4. *Domain analysis*

We examine why the real-to-sim-to-real approach is effective by analyzing the domain of the datasets. From each of the three datasets (Train-real2sim, Train-sim, and Test-OneMAV), 400 images are randomly selected for domain analysis. The t-Distributed Stochastic Neighbor Embedding (t-SNE) [19] is used to visualize the datasets in a two-dimensional map.

The domain analysis results are shown in Fig. 6. As can be seen, there is a significant overlap between datasets Train-real2sim and Test-OneMAV, while there is a little overlap between datasets Train-sim and Test-OneMAV. This explains the effectiveness of the proposed real-to-sim-to-real approach. Furthermore, the distribution of Test-OneMAV clustering is within a broad area. By contrast, the distribution of Train-real2sim is broader and more dispersed. This is because the dataset Test-OneMAV is collected in the real world and can only cover limited areas of the environment. While the dataset Train-real2sim is collected in the simulation and can cover sufficient corners of the environment easily.

4. Estimation and Control

This section introduces the estimation and control algorithms.

4.1. *Bearing and distance measurements*

After the target MAV has been detected in the image, a bounding box that tightly surrounds the target can be obtained. From the bounding box, two types of useful information can be extracted. The first is the unit bearing vector $\hat{g} \in \mathbb{R}^3$ pointing from the camera to the target. It can be calculated based on the center coordinate of the bounding box and the intrinsic parameters of the camera. The second is the angle $\hat{\theta}$ subtended by the target in the camera's field of view. It can be calculated based on the size of the bounding box and the intrinsic parameters of the camera. Detailed calculation can be found in our previous work [4, 20] and is omitted here due to space limitation.

Different from [4, 20], we also incorporate an estimate of the target's physical size denoted as $\ell \in \mathbb{R}$. Here, ℓ is a scalar that represents the physical size of the target MAV in the dimension that is orthogonal to the bearing vector [17]. In this paper, we suppose that that the target can be approximated as a cylinder shape and the physical size refers to the diameter of the cylinder. As a result, ℓ is approximately invariant when the target is viewed from different angles. Since the target is non-cooperative, we do not know its physical size in advance. Denote $\hat{\ell}$ as the estimate of the physical size. It can be easily obtained from the geometry that

$$\hat{r} = \frac{\hat{\ell}/2}{\tan(\hat{\theta}/2)}, \tag{1}$$

where $\hat{r}$ is the distance measurement of the target MAV.

The distance measurement $\hat{r}$ is obtained as follows.

(1) We first select an initial guess of $\hat{\ell}$, based on which we can calculate $\hat{r}$ using (1). Since $\hat{\ell}$ may be inaccurate, $\hat{r}$ obtained in this way may also be inaccurate. However, $\hat{\ell}$ can be further updated to be more accurate as shown below.

May 4, 2024 ISSN: 2301-3850

(2) We have a one-beam laser ranger integrated with the camera. $\hat{r}$ can be directly measured by the laser ranger when the target is aligned with the principal axis of the camera. In this case, the value of $\hat{\ell}$ can be corrected using $\hat{\ell} = 2\hat{r}\tan(\hat{\theta}/2)$, which can be obtained by rewriting (1). The updated value of $\hat{\ell}$ can be further used to calculate $\hat{r}$ using (1) when the laser ranger is ineffective. Note that $\hat{\ell}$ can be corrected whenever the laser ranger is effective.

4.2. *Pseudo-linear Kalman filter*

The conventional Extended Kalman Filter (EKF) exhibits divergence problems [21, 22] when applied to bearing-based target motion estimation, especially in scenarios with significant initial errors [22]. In contrast, the pseudo-linear Kalman filter shows stable performance [4, 17, 23, 24]. In this paper, we adopt the pseudo-linear Kalman filter to estimate the target's motion states.

Denote $p_T, v_T \in \mathbb{R}^3$ as the target's position and velocity, respectively. Let $x = [p_T^{\mathrm{T}}, v_T^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$ be the state vector that we aim to estimate. Suppose that the target's motion can be modeled as a noise-driven double integrator [4, 20, 25]:

$$x(t_{k+1}) = Fx(t_k) + Bq(t_k), \tag{2}$$

where

$$F = \begin{bmatrix} I_{3\times3} & \delta t I_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \in \mathbb{R}^{6\times6}, \quad B = \begin{bmatrix} \frac{1}{2}\delta t^2 I_{3\times3} \\ \delta t I_{3\times3} \end{bmatrix} \in \mathbb{R}^{6\times3}.$$

Here, δt is the sampling time, and I and 0 are the identity and zero matrices, respectively. Moreover, $q \sim \mathcal{N}(0, \Sigma_q) \in \mathbb{R}^3$ is the process noise, whose covariance matrix is $\Sigma_q = \mathrm{diag}(\sigma_a^2, \sigma_a^2, \sigma_a^2) \in \mathbb{R}^{3\times3}$. Here, $\sigma_a \in \mathbb{R}$ is the standard deviation of the target's random acceleration.

The bearing $\hat{g}$ and distance $\hat{r}$ measurements are nonlinear functions of the target's position:

$$\hat{g} = \frac{p_T - p_C}{\|p_T - p_C\|} + \mu, \tag{3}$$

$$\hat{r} = \|p_T - p_C\| + w, \tag{4}$$

where $p_C \in \mathbb{R}^3$ is the position of the onboard camera, and $\mu \sim \mathcal{N}(0, \sigma_\mu^2 I_{3\times3})$ and $w \sim \mathcal{N}(0, \sigma_w^2)$ are measurement noises. By letting $\sigma_w \neq 0$, we can handle the case where ℓ varies slightly. In this paper, the positions of the camera and the catcher MAV are assumed to be the same.

Equations (3) and (4) can be rewritten as pseudo-linear measurement equations [20, Sec. 2.1]:

$$z = Hx + \nu, \tag{5}$$

where

$$z = \begin{bmatrix} P_{\hat{g}} p_C \\ p_C + \hat{r}\hat{g} \end{bmatrix} \in \mathbb{R}^6, \quad H = \begin{bmatrix} P_{\hat{g}} & 0_{3\times3} \\ I_{3\times3} & 0_{3\times3} \end{bmatrix} \in \mathbb{R}^{6\times6},$$

$$\nu = \begin{bmatrix} rP_{\hat{g}}\mu \\ r\mu + w\hat{g} \end{bmatrix} \in \mathbb{R}^6,$$

where $P_{\hat{g}} \doteq I_{3\times3} - \hat{g}\hat{g}^{\mathrm{T}} \in \mathbb{R}^{3\times3}$ is an orthogonal projection matrix [4]. Here, ν can be treated as a linear combination of Gaussian noises, and hence handled by the Kalman filter [4, 23, 24, 26]. Its covariance matrix $\Sigma_\nu \in \mathbb{R}^{6\times6}$ can be calculated as

$$\Sigma_\nu = \begin{bmatrix} rP_{\hat{g}} & 0_{3\times1} \\ rI_{3\times3} & \hat{g} \end{bmatrix} \begin{bmatrix} \sigma_\mu^2 I_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & \sigma_w^2 \end{bmatrix} \begin{bmatrix} rP_{\hat{g}} & 0_{3\times1} \\ rI_{3\times3} & \hat{g} \end{bmatrix}^{\mathrm{T}},$$

where r can be further replaced by $\hat{r}$ [4, 27].

With the state transition equation (2) and the pseudo-linear measurement equation (5), the target motion estimator can be realized by the Kalman filter.

4.3. *Control algorithms*

The control objective is that the catcher MAV should maintain desired separations in both the horizontal and vertical directions. In particular, the desired horizontal separation is $d_h = 4$ and the desired vertical separation is $d_v = 3$ (Fig. 8(c)). Moreover, the yaw angle of the catcher MAV should be controlled so that the net launcher installed underneath the catcher MAV is aligned with the target MAV. Given that the system is fully automatic, a set of net launching conditions is required to ensure that the net launcher is aimed at the target during the launching process.

For the horizontal separation, we design the following velocity command, which is further tracked by a low-level flight controller. In particular, let $p_{C_h} \in \mathbb{R}^2$ be the catcher's horizontal position, and $p_{T_h}, v_{T_h} \in \mathbb{R}^2$, be the estimated target's horizontal position and velocity, respectively. Then, $r_h = \|p_{T_h} - p_{C_h}\|_2$ is the horizontal distance between the two MAVs. The horizontal velocity control command is

$$v_{C_h}^{\mathrm{cmd}} = v_{T_h} + k_h \frac{r_h^2 - d_h^2}{r_h^3}(p_{T_h} - p_{C_h}),$$

where $k_h \in \mathbb{R}$ is a positive coefficient.This control command is designed as a gradient-descent algorithm that can minimize the Lyapunov energy function [18] $(r_h - d_h)^2/r_h$.

For the vertical separation, we simply set the desired position of the catcher MAV as $p_{C_v}^{\mathrm{cmd}} = p_{T_v} + d_v$, which is further tracked by the low-level flight controller. Here, $p_{T_v} \in \mathbb{R}$ is the estimated height of the target MAV.

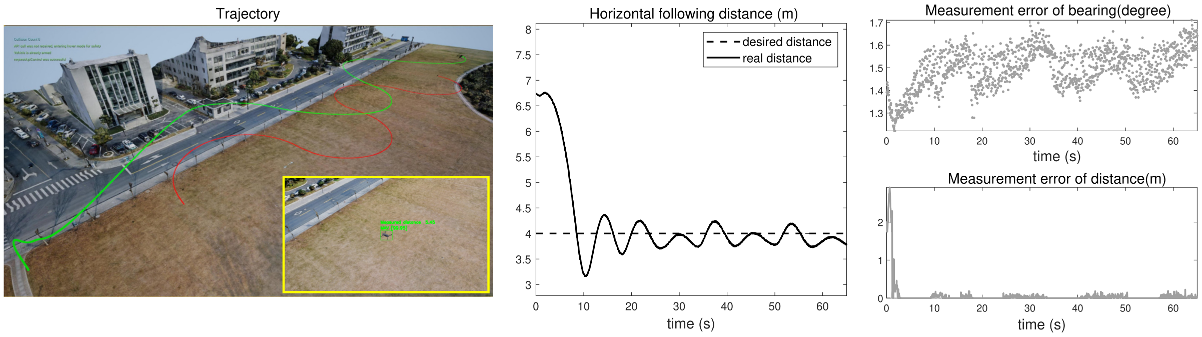


Fig. 7. Experimental results of the AirSim simulation. The catcher MAV successfully pursues the target.

For the yaw angle control, we set the desired yaw angle of the catch MAV as the yaw angle of the gimbal camera: $\psi_C^{\text{cmd}} = \psi_{\text{gim}}$. Then, the yaw angle command is tracked by low-level flight control of the catcher MAV. Since the gimbal camera is controlled to keep the target at the center of the camera's field of view, this control command can align the heading of the catcher MAV (and also the net launcher) with the target.

For the net launching conditions, we propose three conditions based on outdoor experimental experience. The effective range for the net launcher is 2–8 m. We choose 5 m as the best catching distance. The net launcher is installed with a pitch angle of 35° (see Fig. 8(c)). According to the above analysis, we propose three net launching conditions:

(1) The catcher's heading is towards the target;
(2) The catcher MAV's pitch and roll angle is less than 1°, which means that the catcher tracks the target stably;
(3) The catcher is 3 m higher than the target, and 4 m away from the target in the horizontal plane.

For safety reasons, in our real-world experiments, the above conditions must be consistently met for 2 sec before launching.

5. AirSim Simulation

In this section, we test the MAV-catching-MAV system in the AirSim simulation environment. All the vision, estimation, and control algorithms are incorporated.

5.1. *Simulation setup*

The simulation environment is constructed from real-world images (see Fig. 1). The catcher and the target MAV models are selected as DJI M300 and Mavic2, respectively. The catcher MAV can capture images using its simulated onboard camera (left subfigure of Fig. 7). The MAV detector is a tiny-Yolo v4 network that is trained based on the dataset Train-Real2Sim. A simulated laser ranger is also incorporated into the simulation.

The target MAV moves along an S-shaped trajectory (5 m of radius) with a speed of 1 m/s (see the red curve in the left subfigure of Fig. 7). This scenario is challenging because the target maneuvers aggressively. The catcher's maximum speed is limited to 2 m/s. Parameters in the estimator are selected as $\sigma_a^2 = 1$, $\sigma_\mu^2 = 10^{-4}$ and $\sigma_w^2 = 0.1$. The initial value of the target's physical size is set to $\hat{\ell} = 0.25$, which is different from the ground-truth value $\ell = 0.354$. We use the same algorithm parameter values in real-world experiments. The initial distance between the two MAVs is set to be seven.

5.2. *Simulation results*

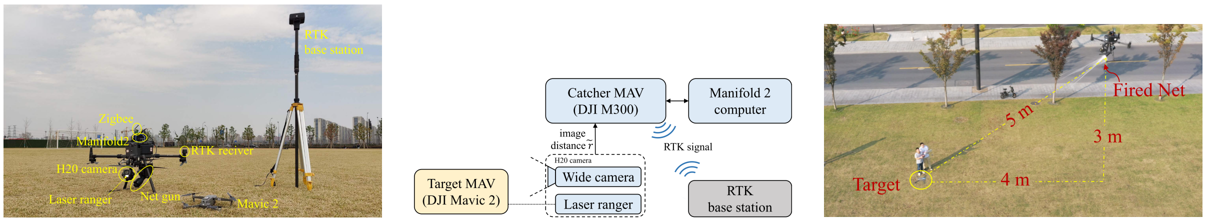
Figure 7 shows the simulation results. As can be seen, the catcher MAV successfully pursues the target, and the error between the desired and true relative distance oscillates within 0.5. The error cannot converge to zero because the target maneuvers and hence the estimation has errors. The trajectories of two MAVs are shown in the left subfigure of Fig. 7.

The measurement errors are shown in the right subfigure in Fig. 7. As can be seen, the error of bearing measurement is inversely correlated to the true distance between the target and the catcher. This is reasonable because, when the target is close to the camera hence the size of the bounding box is large, the center point of the bounding box usually varies for a few pixels [17]. Moreover, the error of distance measurement is large at the beginning of the simulation and reduces after the target's physical size is calibrated.

6. Real-World Experiments

In this section, real-world experiments are presented to further verify the effectiveness of the proposed system.

The hardware platform is shown in Fig. 8(a). The catcher MAV is developed based on a DJI M300 quadcopter which



(a) Hardware platforms.

(b) Hardware communication structure.

(c) Net launching conditions.

Fig. 8. The hardware of the MAV-catching-MAV system.

can accurately self-localize using an RTK GPS. The catcher MAV has a monocular gimbal camera (H20). The resolution of images obtained from this camera is 1920 × 1080 pixels. We compress them to 1536 × 864 pixels for the detection process. A single-beam laser ranger is integrated with the camera and can provide distance when the target appears at the center of the image. Key specifications of the catcher MAV and its onboard equipments are listed in Table 3. All proposed algorithms are deployed on the onboard Manfold 2G computer and executed online in real-time. The frequency of the vision detection is 5 Hz. The frequency of both the estimation and control nodes is 50 Hz. The parameters of the algorithms are the same as those used in the AirSim simulation. The communication between the onboard systems is shown in Fig. 8(b). We conducted two sets of experiments, the details of which are given as follows.

6.1. *Experiment 1: MAV-following-MAV*

In experiment 1, we use another DJI M300 quadcopter as the target. The task is that the catcher MAV should follow the target MAV with the desired separation. Here, the target is controlled manually, moving along an approximately straight line with a speed around 1 m/s. The experimental results are shown in Fig. 9(a). Although the DJI M300 MAV has not been seen in the training dataset, the MAV detector works effectively in this case. Moreover, the overall system works effectively. The catcher MAV can successfully follow the target MAV and maintain the desired separation.

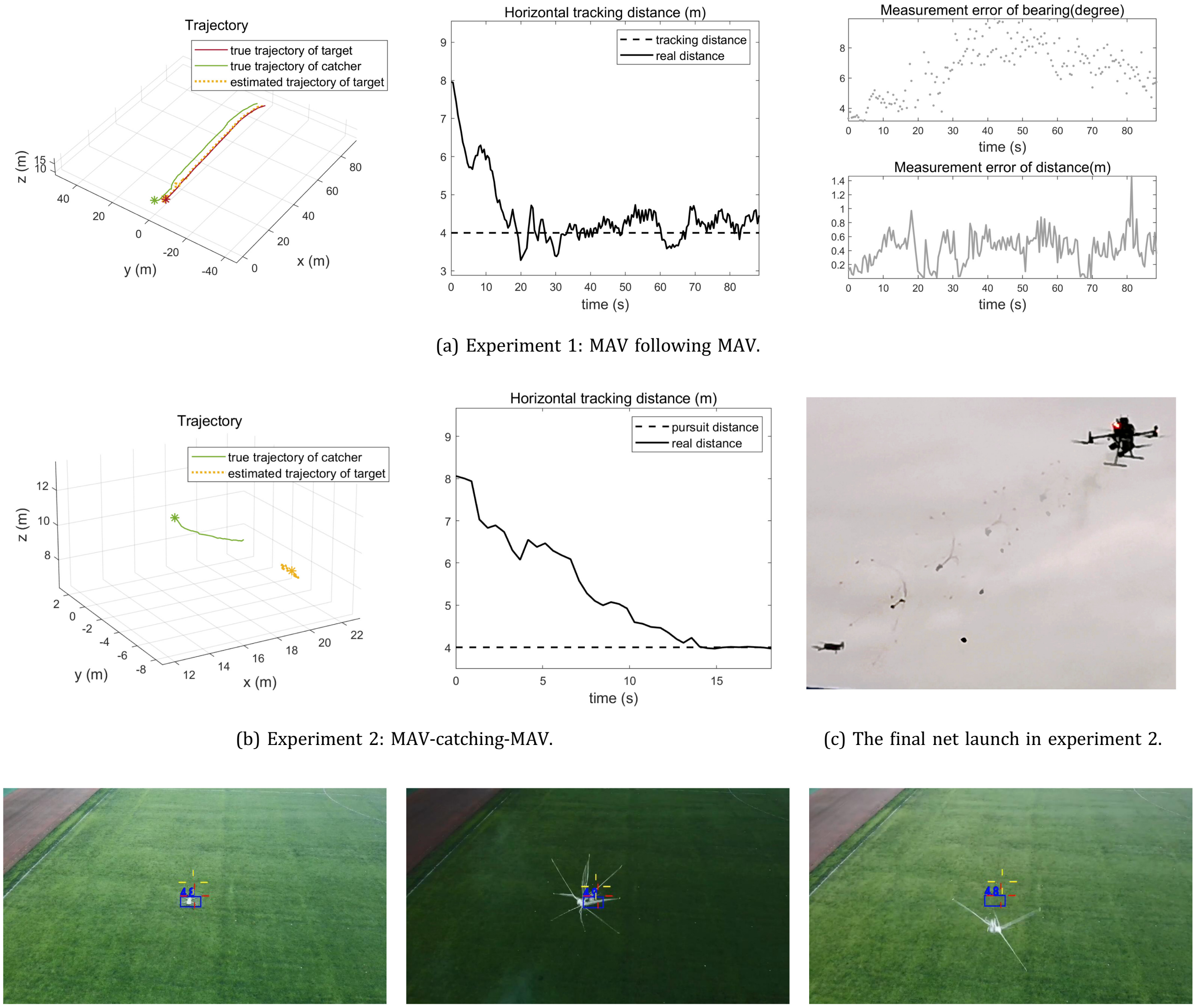
6.2. *Experiment 2: MAV-catching-MAV*

In experiment 2, we use a DJI Mavic2 quadcopter as the target. The task is that the catcher MAV should maintain a desired relative position and then launch a net to capture the target. Although the target MAV is commanded to hover, it still moves at low speed due to inaccurate GPS and wind disturbance.

The experimental results are shown in Fig 9(b). As can be seen, the catcher MAV can effectively maintain a desired relative position. Figures 9(c) and 9(d) show the time instance when the net is flying to catch the target MAV. The photos are taken from the ground and onboard views, respectively. As can be seen, the target can be successfully caught.

Table 3. Key specifications of the hardware platform.

	Parameter	Value	Unit
M300 quadcopter	Diagonal size	895	mm
	Total mass	7.4	kg
	Max takeoff weight	9	kg
	Max flight time	30	min
RTK	Accuracy	1	cm
	Max frequency	10	Hz
H20 camera & gimbal	Resolution	1920×1080	pixel
	Frequency	15	Hz
	Max angular rate	180	deg/s
Laser ranger	Range	3–1200	m
	Accuracy	0.2–1.7	m
Net launcher	Range	2–10	m
	Max coverage area	5×5	m^2
Mavic 2	Diagonal size	354	mm
	Total mass	0.9	kg



(a) Experiment 1: MAV following MAV.

(b) Experiment 2: MAV-catching-MAV.

(c) The final net launch in experiment 2.

(d) The onboard camera's view in the final net launch procedure in experiment 2.

Fig. 9. The results of the outdoor experiments.

We conducted more than 20 flight experiments. Only three tests failed. Two of them were caused by the mechanical failure of the net launcher. The other was caused by strong wind gusts that deviated the flying net from its intended trajectory.

7. Conclusion

This paper proposed an autonomous drone-catching-drone system, where a catcher MAV can detect, estimate, track and catch a non-cooperative target MAV automatically. One core contribution is that we proposed a real-to-sim-to-real approach that can effectively alleviate the sim-to-real gap. Both simulation and real-world experiments verified the effectiveness of the proposed approach and system.

ORCID

Zian Ning https://orcid.org/0000-0003-1784-6910
Yin Zhang https://orcid.org/0000-0002-9347-0241
Xiaofeng Lin https://orcid.org/0009-0006-5180-2232
Shiyu Zhao https://orcid.org/0000-0003-3098-8059

References

[1] G. K. Taylor, A. L. Thomas and C. H. Brighton, Air-to-air and air-to-ground attack strategies in trained birds of prey, Technical report, University of Oxford Department of Zoology (2014).

[2] M. Vrba and M. Saska, Marker-less micro aerial vehicle detection and localization using convolutional neural networks, *IEEE Robot. Autom. Lett.* **5** (2020) 2459–2466.

[3] C. Dong, C. Liu, J. Liu, N. Zhou, H. Zhang and T. Fang, An integrated scheme of a smart net capturer for muavs, *IEEE Access* **8** (2020) 211820–211828.

[4] J. Li, Z. Ning, S. He, C.-H. Lee and S. Zhao, Three-dimensional bearing-only target following via observability-enhanced helical guidance, *IEEE Trans. Robot.* **39** (2023) 1509–1526.

[5] 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.* **6** (2021) 1020–1027.

[6] Y. Zheng, C. Zheng, X. Zhang, F. Chen, Z. Chen and S. Zhao, Detection, localization, and tracking of multiple MAVs with panoramic stereo camera networks, *IEEE Trans. Autom. Sci. Eng.* **20** (2023) 1226–1243.

[7] V. Walter, M. Vrba and M. Saska, On training datasets for machine learning-based visual relative localization of micro-scale UAVs, in *2020 IEEE Int. Conf. Robotics and Automation (ICRA)*, 2020, pp. 10674–10680.

[8] D. Marez, S. Borden and L. Nans, UAV detection with a dataset augmented by domain randomization, in *Geospatial Informatics X*, eds. K. Palaniappan, G. Seetharaman, P. J. Doucette and J. D. Harguess (SPIE, 2020).

[9] T. R. Dieter, A. Weinmann, S. Jäger and E. Brucherseifer, Quantifying the simulation reality gap for deep learning-based drone detection, *Electronics* **12** (2023) 2197.

[10] C. Rui, G. Youwei, Z. Huafei and J. Hongyu, A comprehensive approach for uav small object detection with simulation-based transfer learning and adaptive fusion, arXiv, preprint (2021), arXiv:2109.01800.

[11] E. Cetin, C. Barrado and E. Pastor, Improving real-time drone detection for counter-drone systems, *Aeronaut. J.* **125** (2021) 1871–1896.

[12] C. Briese and L. Guenther, Deep learning with semi-synthetic training images for detection of non-cooperative UAVs, *2019 Int. Conf. Unmanned Aircraft Systems (ICUAS)*, IEEE, 2019, pp. 981–988.

[13] A. Barisic, F. Petric and S. Bogdan, Sim2air — synthetic aerial dataset for uav monitoring, *IEEE Robot. Autom. Lett.* **7** (2022) 3757–3764.

[14] C. Symeonidis, C. Anastasiadis and N. Nikolaidis, A UAV video data generation framework for improved robustness of uav detection methods, *2022 IEEE 24th Int. Workshop on Multimedia Signal Processing (MMSP)*, 2022, pp. 1–5.

[15] J. Rothe, M. Strohmeier and S. Montenegro, A concept for catching drones with a net carried by cooperative UAVs, *2019 IEEE Int. Symp. Safety, Security, and Rescue Robotics (SSRR)*, 2019, pp. 126–132.

[16] F. Yuan, C. Liu, C. Dong, S. Wang and B. Ye, Integrated design and research on detection-guidance-control of anti-micro uav system, *6th Int. Conf. Automation, Control and Robots (ICACR)*, IEEE, 2022, pp. 126–130.

[17] Z. Ning, Y. Zhang, J. Li, Z. Chen and S. Zhao, A bearing-angle approach for unknown target motion analysis based on visual measurements, *Int. J. Robot. Res.* (2024) 1–20.

[18] A. M. Lyapunov, The general problem of the stability of motion, *Int. J. Control* **55** (1992) 531–534.

[19] L. van der Maaten and G. Hinton, Visualizing data using t-sne, *J. Mach. Learn. Res.* **9**(86) (2008) 2579–2605.

[20] Z. Ning, Y. Zhang and S. Zhao, Comparison of different pseudo-linear estimators for vision-based target motion estimation, *Control Theory Technol.* **21** (2023) 448–457.

[21] V. Aidala, Kalman filter behavior in bearings-only tracking applications, *IEEE Trans. Aerosp. Electron. Syst.* **AES-15** (1979) 29–39.

[22] X. Lin, T. Kirubarajan, Y. Bar-Shalom and S. Maskell, Comparison of EKF, pseudomeasurement, and particle filters for a bearing-only target tracking problem, in *SPIE Proc.*, ed. O. E. Drummond (SPIE, 2002).

[23] A. Lingren and K. Gong, Position and velocity estimation via bearing observations, *IEEE Trans. Aerosp. Electron. Syst.* **AES-14** (1978) 564–577.

[24] V. Aidala and S. Nardone, Biased estimation properties of the pseudolinear tracking filter, *IEEE Trans. Aerosp. Electron. Syst.* **AES-18** (1982) 432–441.

[25] S. He, H.-S. Shin and A. Tsourdos, Trajectory optimization for target localization with bearing-only measurement, *IEEE Trans. Robot.* **35** (2019) 653–668.

[26] K. Dogancay, 3D pseudolinear target motion analysis from angle measurements, *IEEE Trans. Signal Process.* **63** (2015) 1570–1580.

[27] S. He, J. Wang and D. Lin, Three-dimensional bias-compensation pseudomeasurement kalman filter for bearing-only measurement, *J. Guid. Control Dyn.* **41** (2018) 2678–2686.

Zian Ning received the B.S. degree in electrical engineering from Northwestern Polytechnical University, Xi'an, China, in 2015, and the M.S. degree in aeronautical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently working toward the Ph.D. degree in the Intelligent Unmanned Systems Laboratory at Westlake University, Hangzhou, China. His research interests include vision-based multi-robot cooperation, state estimation, and filtering.

Yin Zhang received the B.S. degree in measurement and control technology and instrumentation from the Tianjin University, Tianjin, China, in 2017, and the M.S. degree in instrument science and technology from Beihang University, Beijing, China, in 2020. She is currently working toward the Ph.D. degree in the Intelligent Unmanned Systems Laboratory at Westlake University, Hangzhou, China. Her research interests include domain adaptation, drone detection and depth estimation.

Xiaofeng Lin received B.Eng. in Engineering Mechanics from Tianjin University, China, in 2020, and MS in Robotics from University of Michigan, Ann Arbor in 2023.

He is currently a first-year Ph.D. student at Boston University. His research interests lie in the intersection of online learning, reinforcement learning and multi-agent systems.

Shiyu Zhao received the B.Eng. and M.Eng. degrees from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2006 and 2009, respectively, and the Ph.D. degree from the National University of Singapore, Singapore, in 2014, all in electrical engineering.

From 2014 to 2016, he was a Postdoctoral Researcher at the Technion-Israel Institute of Technology, Haifa, Israel, and the University of California at Riverside, Riverside, CA, USA. From 2016 to 2018, he was a Lecturer in the Department of Automatic Control and Systems Engineering at the University of Sheffield, Sheffield, UK. He is currently an Associate Professor in the School of Engineering at Westlake University, Hangzhou, China. His research interests lie in sensing, estimation, and control of robotic systems.