

Vision-based Relative Detection and Tracking for Teams of Micro Aerial Vehicles

Rundong Ge^{1*}, Moonyoung Lee^{1*}, Vivek Radhakrishnan^{1,2}, Yang Zhou¹, Guanrui Li¹, and Giuseppe Loianno¹

Abstract—In this paper, we address the vision-based detection and tracking problems of multiple aerial vehicles using a single camera and Inertial Measurement Unit (IMU) as well as the corresponding perception consensus problem (i.e., uniqueness and identical IDs across all observing agents). We design several vision-based decentralized Bayesian multi-tracking filtering strategies to resolve the association between the incoming unsorted measurements obtained by a visual detector algorithm and the tracked agents. We compare their accuracy in different operating conditions as well as their scalability according to the number of agents in the team. This analysis provides useful insights about the most appropriate design choice for the given task. We further show that the proposed perception and inference pipeline which includes a Deep Neural Network (DNN) as visual target detector is lightweight and capable of concurrently running control and planning with Size, Weight, and Power (SWaP) constrained robots on-board. Experimental results show the effective tracking of multiple drones in various challenging scenarios such as heavy occlusions.

SUPPLEMENTARY MATERIAL

Video: <https://youtu.be/mv5F7NLue6A>

Code: https://github.com/arplaboratory/multi_robot_tracking

I. INTRODUCTION

Teams of Micro Aerial Vehicles (MAVs), often called swarms for large team sizes, are emerging as a disruptive on-demand technology to deploy distributed and intelligent autonomous systems [1] for environment coverage, monitoring, situational awareness, transportation, and communication (e.g., creation of ad hoc remote networks). Application areas include but are not limited to agriculture, search and rescue, inspection, public safety (e.g., COVID-19 monitoring), warehouse management, and entertainment. By enabling cooperation among agents, multiple aerial vehicles offer additional flexibility, resilience, and robustness in several tasks compared to a single robot [2]. However, several challenges remain in order to autonomously deploy them in real-world scenarios. To enable high-level autonomous decision-

*These authors contributed equally.

¹The authors are with the New York University, Tandon School of Engineering, Brooklyn, NY 11201, USA. email: {rundong.ge, m17617, vr2171, yangzhou, lguanrui, loianno}@nyu.edu.

²The author is with the Technology Innovation Institute, Abu Dhabi, UAE. email: vivek.radhakrishnan@tii.ae.

This work was supported by the NSF CAREER Award 2145277, the NSF CPS Grant CNS-2121391, the Technology Innovation Institute, Qualcomm Research, Nokia, and NYU Wireless. Giuseppe Loianno serves as consultant for the Technology Innovation Institute. This arrangement has been reviewed and approved by the New York University in accordance with its policy on objectivity in research.

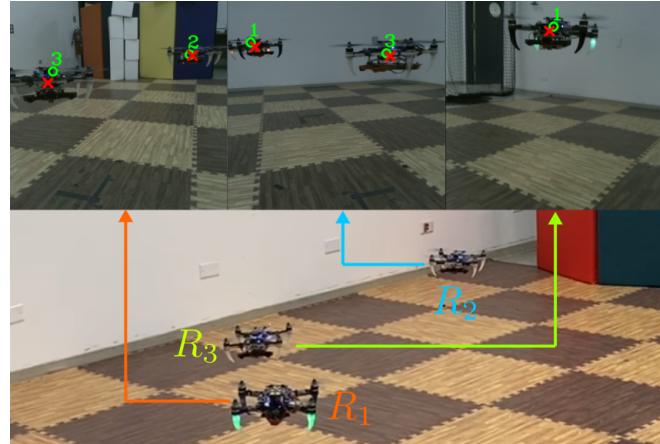


Fig. 1: Multi-target tracking in each of the MAV's perspective. Tracked agents are in consensus among the three MAVs (green dots) using unsorted measurements (red crosses).

making policies, agents require on-board self-localization with respect to other robots [1], [2] in a decentralized fashion with minimal communication. In this work, we address the vision-based decentralized detection and tracking for multiple MAVs with perception consensus among the agents as shown in Fig. 1. With a deep consideration for deploying MAVs constrained by Size, Weight, and Power (SWaP), we designed a lightweight perception pipeline that rely only on a single camera and Inertial Measurement Unit (IMU). These sensors have become popular due to low energy requirements of small-scale robots. Early works on formation control [3]–[5] rely on an external motion capture system to detect and track the agents. Our previous works [6], [7] address the on-board control and planning to generate collision-free formation trajectories in obstacle-free environments. However, the approach is not decentralized since the swarm framework relies on a ground station to resolve the relative detection and perception consensus. This latest work deploys team of MAVs without any central framework but instead only requires initial fleet configuration knowledge a priori. The drone-to-drone relative localization has generally been addressed by additionally mounting artificial visual markers [8]–[13] or employing a multi-modal sensor fusion approaches [14]–[16], including Ultra-Wide Band (UWB) which is the main focus in [16]. However, UWB technology still requires the installation of a specific system infrastructure.

Conversely, other works [17]–[20] address the vision-

based tracking problem. However, [17] does not focus on multiple vehicles. In [18], the authors address the relative localization in a centralized manner. The approach relies on a powerful ground station to resolve localization and tracking. Moreover, only simulation results and evaluations on datasets are presented. In [19], the tracking problem is resolved without explicit perception consensus among agents as well as bypassing the challenges related to deploying these approaches onto SWaP constrained robots. Furthermore, similar to [21] the authors employ only a specific filtering solution without comparative analysis. Finally, our previous work [20] only considered the detection and tracking problems in a centralized manner from a VR-headset. Conversely, in [22] the authors compare several tracking filtering solutions, but only provide simulation results and do not address the specificity of vision-based multi-tracking problem in different conditions neither the perception consensus. They also do not address the real-time deployment on SWaP constrained robots. Other works [23], [24] achieve indirect relative localization by sharing a set of characteristic landmarks across the agents. Although these works relax the line-of-sight requirements, they require communication among the agents to share and store local maps. Recently, learning-based solutions for multi-target tracking have also surged in popularity [25]. However, these methods are computationally expensive to run on SWaP constrained robots.

The contributions of this paper are twofold. First, we design several decentralized Bayesian vision-based multi-tracking filtering strategies to resolve the association problem between the incoming unsorted measurements obtained from a visual detector with perception consensus such that all observing drones agree on tracking targets' IDs. We compare their accuracy in different operating conditions as well as their scalability according to the number of agents in the team. Second, we show how the proposed setup including a Deep Neural Network (DNN) acting as visual target detector is able to run on-board in real-time on a small fleet of SWaP constraint MAVs concurrently with planning and control. The proposed pipeline utilizes only a minimalistic sensor suite composed of a single camera and IMU. To the best of our knowledge, this work shows the first vision-based detection and tracking for multiple MAVs in a decentralized manner where each agent is only equipped with a single camera and IMU while concurrently running planning and control in real-time. Overall, this approach can be deployed on-demand without relying on any external infrastructure or marker with the potential to scale to swarms of aerial robots.

The paper is organized as follows. Section II describes the proposed approach. Section III analyzes the accuracy, computational complexity, and scalability of the methods. Section IV presents the experimental results, Section V discusses the results, and Section VI concludes the paper.

II. METHODOLOGY

A. Preliminaries

We consider a system of robots equipped with a camera and an IMU. Without loss of generality, we assume that

for each robot, the camera and IMU frames are coincident with the robot frame. An external calibration procedure can compute the relative transformation between the frames. Our algorithm provides the state of each tracked agent in each robot or camera frame. We focus on the tracking problem without considering the relative pose problem (i.e., the estimated pose of each agent) that can be solved in parallel as in [20], [26]. In the following, we describe the multi-tracking procedure from an observing drone of a generic agent i . We design and analyze three Bayesian filters most representatives of several multi-target tracking filtering categories. Specifically, we design a Kalman Filter with the maximum likelihood of association (unimodal approach), a Joint Probabilistic Association filter (explicit computation of all possible association), and a Probability Hypothesis Density filter (random finite sets). These differ in the way each incoming j^{th} measurement $\mathbf{z}_k^j \in Z_k$ obtained in the camera frame at time k from a visual target detector (e.g., a DNN in our settings in Section IV) is associated with an existing tracked agent \mathbf{x}_k^i with $i \in \{1 \dots n\}$. We denote the j^{th} measurement associated with a tracked agent \mathbf{x}_k^i as $\mathbf{z}_k^{i,j} \in Z_k$. In the filters, the IMU is used for the filter prediction whereas the visual measurements as update.

Each agent i is tracked directly in the image plane using a 4-dimensional vector which contain the track positions and velocities along the u and v image axes

$$\mathbf{x}_k^i = [p_u^i \quad \dot{p}_u^i \quad p_v^i \quad \dot{p}_v^i]^\top. \quad (1)$$

The relative motion between observed and target drones can then be represented by a stochastic nonlinear differential equation with a constant speed motion model

$$\mathbf{x}_{k+1}^i = f(\mathbf{x}_k^i, \mathbf{u}_k, q) = \mathbf{A}_k \mathbf{x}_k^i + \mathbf{B}_k(\mathbf{x}_k^i) \mathbf{u}_k + \mathbf{Q}_k, \quad (2)$$

$$\mathbf{A}_k = \begin{bmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q}_k = q \begin{bmatrix} \frac{\delta t^2}{2} & 0 & 0 & 0 \\ 0 & \delta t & 0 & 0 \\ 0 & 0 & \frac{\delta t^2}{2} & 0 \\ 0 & 0 & 0 & \delta \end{bmatrix} \quad (3)$$

$$\mathbf{B}_k = \delta t \begin{bmatrix} \frac{(p_u^i - c_u)(p_v^i - c_v)}{f} & -\frac{(p_u^i - c_u)^2}{f} - f & p_v^i - c_v \\ 0 & 0 & 0 \\ f + \frac{(p_v^i - c_v)^2}{f} & -\frac{(p_u^i - c_u)(p_v^i - c_v)}{f} & -p_u^i + c_u \\ 0 & 0 & 0 \end{bmatrix} \quad (4)$$

where \mathbf{u}_k is the angular velocity provided by the IMU in the robot frame of the observing agent, \mathbf{A}_k is the state transition matrix, \mathbf{B}_k is derived from the optical flow equation, and \mathbf{Q}_k is the process noise covariance matrix. Specifically, δt is the sampling time at frame k , q is the acceleration of the drones in px/s^2 assumed to be a Gaussian random variable, (c_u, c_v) are the principal point coordinates, f is the focal length. The reader can refer to [20] for more details of this model.

B. Multi-Target Tracking

1) *Kalman Filter:* At every iteration of the Kalman filter, the algorithm includes a prediction and update steps.

Prediction step: In this step, the robot computes the predicted state of each target using classic Kalman filter equations for the predicted mean and covariance based on the motion model equation defined in eq. (2).

Update step: In the update step, before incorporating the measurement information for each tracked agent, it is necessary to associate each tracked agent \mathbf{x}_k^i with a given measurement j to apply the Kalman filter update equations obtained as updated mean and covariance respectively

$$\begin{aligned}\boldsymbol{\mu}_{k|k}^i &= \boldsymbol{\mu}_{k|k-1}^i + \mathbf{K}_k \mathbf{y}_k^i, \\ \mathbf{P}_{k|k-1}^i &= \mathbf{F}_{k-1} \mathbf{P}_{k-1}^i \mathbf{F}_{k-1}^\top + \mathbf{Q}_{k-1}, \\ \mathbf{F}_k &= \mathbf{A}_k + \frac{\partial \mathbf{B}_k}{\partial \mathbf{x}_k^i} \mathbf{u}_k,\end{aligned}\quad (5)$$

where $\mathbf{y}_k^i = \mathbf{z}_k^{i,j} - \mathbf{H}_k \boldsymbol{\mu}_{k|k-1}^i$ is the innovation term, \mathbf{K}_k^i is the Kalman gain, $\boldsymbol{\mu}_{k|k-1}^i$ is the predicted mean of the state \mathbf{x}_k^i at the prediction step, \mathbf{H}_k is the measurement model defined in [20]. The association probability p of the tracked agent i with a measurement j at time k denoted as $\beta_k^{i,j}$ is obtained according to [27] by selecting the maximum posterior distribution with respect to $\mathbf{x}_k^i, \mathbf{z}_k^j$ pair as

$$\begin{aligned}p(\beta_k^{i,j} | \mathbf{z}_k^j, \mathbf{x}_k^i) &\propto p(\mathbf{z}_k^j | \beta_k^{i,j}, \mathbf{x}_k^i) p(\beta_k^{i,j}) = \\ \mathcal{N}(\mathbf{z}_k^j - \boldsymbol{\mu}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) p(\beta_k^{i,j})\end{aligned}\quad (6)$$

where \mathcal{N} is a normal distribution.

2) *Joint Probabilistic Data Association Filter:* The JPDAF is also divided into a prediction and update steps.

Prediction step: In this step, the robot computes the predicted state in the same way as in the Kalman filter case.

Update step: Similar to the Kalman filter, before incorporating the measurement information, the JPDAF explicitly resolves the association problem by computing all possible associations between the tracked agents and the incoming measurements. These associations are represented in a matrix form as β_k^i . For each agent i , the update is performed using eq. (5), considering

$$\begin{aligned}\mathbf{y}_k^i &= \sum_{j=1}^{Z_k} \beta_k^{i,j} (\mathbf{z}_k^{i,j} - \mathbf{H}_k \boldsymbol{\mu}_{k|k-1}^i), \\ \beta_k^i &= \sum_{\forall \chi} P\{\chi | Z_k\} \cdot \mathbf{I}(\chi)\end{aligned}\quad (7)$$

where $P\{\chi | Z_k\}$ the probability corresponding to each hypothesis matrix for the event χ and measurements Z_k , and $\mathbf{I}(\chi)$ the hypothesis matrix, $\beta_k^{i,j}$ is the j^{th} column of β_k^i . While there are heuristics to reduce the computational burden associated with the explicit computation of all possible associations, such as proximity threshold, generating a hypothesis matrix for all possibilities can be computationally challenging once the number of tracked agents increases. the filter's computational complexity is discussed in Section III.

3) *Gaussian Mixture PHD Filter:* The GM-PHD is divided into a prediction and update step. It represents the measurement \mathbf{z}_k^j and agent state \mathbf{x}_k^i using Random Finite Sets (RFS) instead of explicitly associating all possible matches

between the tracks and measurements. The tracked states can then be represented with density functions over the state space of targets, where the GM-PHD describes the first moment of distribution over the RFS. Each tracked agent state \mathbf{x}_k^i can be described as a single intensity v_k^i consisting of a weighted sum of Gaussian components in the form

$$v_k^i(\mathbf{x}_k^i) = \sum_{l=1}^{J_k} w_k^l \mathcal{N}(\mathbf{x}_k^i; \boldsymbol{\mu}_k^l, \mathbf{P}_k^l), \quad (8)$$

where the Gaussian components \mathcal{N} for the state \mathbf{x}_k^i is characterized by the weight w_k^l , mean $\boldsymbol{\mu}_k^l$, state covariance \mathbf{P}_k^l , with J_k the number of tracked agents. Given the measurement and targets' previous states, the Gaussian components are propagated through the prediction and update steps.

Prediction step: Each agent state i is still described as RFS

$$v_{k|k-1}^i(\mathbf{x}_k^i) = \sum_{l=1}^{J_k} w_{k|k-1}^l \mathcal{N}(\mathbf{x}_k^i; \boldsymbol{\mu}_{k|k-1}^l, \mathbf{P}_{k|k-1}^l) + \gamma(\mathbf{x}_k^i), \quad (9)$$

where J_k is the number of tracked agents of the previous iteration and the corresponding Gaussian components adhere to the same motion model discussed in eq. (2) with

$$\begin{aligned}w_{k|k-1}^l &= p_s w_{k-1}^l, \\ \boldsymbol{\mu}_{k|k-1}^l &= \mathbf{A}_{k-1} \boldsymbol{\mu}_{k-1}^l + \mathbf{B}_{k-1} \mathbf{u}_{k-1} + \mathbf{Q}_{k-1}, \\ \mathbf{P}_{k|k-1}^l &= \mathbf{F}_{k-1} \mathbf{P}_{k-1}^l \mathbf{F}_{k-1}^\top + \mathbf{Q}_{k-1}, \quad \mathbf{F}_k = \mathbf{A}_k + \frac{\partial \mathbf{B}_k}{\partial \mathbf{x}_k^i} \mathbf{u}_k.\end{aligned}\quad (10)$$

Similarly to [19], we also assume an adaptive agent birth model $\gamma(\mathbf{x}_k^i)$ particular to the PHD filter in which new Gaussian components are characterized by w_γ^i , mean $\boldsymbol{\mu}_\gamma^i$, and covariance \mathbf{P}_γ^i . We do not have to account for association problem prior to new measurements, thus each agent's state can be set equal to the mean in the prediction step. We set the probability of survival p_s of the target to 1 since we assume the detected drones remain throughout the experiment.

Update step: Each state RFS is being updated as

$$\begin{aligned}v_k(\mathbf{x}_k^i) &= (1 - p_d) v_{k|k-1}(\mathbf{x}_k^i) \\ &+ \sum_{j=1}^{Z_k} \sum_{l=1}^{J_k} w_k^{j,l} \mathcal{N}(\mathbf{z}_k^j; \boldsymbol{\mu}_{k|k}^l, \mathbf{P}_{k|k}^l),\end{aligned}\quad (11)$$

where p_d is the probability of detection. The weight, mean, covariance, and Kalman gain updates are respectively

$$\begin{aligned}w_k^{j,l} &= \frac{p_d w_{k|k-1}^l q_k^l(\mathbf{z}_k^j)}{\kappa_k + \sum_{l=1}^{J_k} p_d w_{k|k-1}^l q_k^l(\mathbf{z}_k^j)}, \\ \boldsymbol{\mu}_{k|k}^l &= \boldsymbol{\mu}_{k|k-1}^l + \mathbf{K}_k^l (\mathbf{z}_k^j - \mathbf{H}_k \boldsymbol{\mu}_{k|k-1}^l), \\ \mathbf{P}_{k|k}^l &= (\mathbf{I} - \mathbf{K}_k^l \mathbf{H}_k) \mathbf{P}_{k|k-1}^l, \\ \mathbf{K}_k^l &= \mathbf{P}_{k|k-1}^l \mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_{k|k-1}^l \mathbf{H}_k^\top + \mathbf{R}_k)^{-1},\end{aligned}\quad (12)$$

where the clutter or the false positive term κ_k is modeled as a random uniform distribution within the agent's field of view [28], \mathbf{R}_k is the measurement noise covariance, and

Algorithm 1 Perception Consensus

Input: local 2D track[N] T , init 3D pose[N] P
Output: global 2D track[N] ID

- 1: **for** tracked targets t **do**
- 2: **for** received init pose i **do**
- 3: $ProjP[i] = (u_p(i), v_p(i)) \leftarrow Project2D(P[i])$
- 4: $\Delta p(t, i) \leftarrow \| (T[t] - ProjP[i]) \|_2$
- 5: **end for**
- 6: $[val, index] \leftarrow \text{find minimum}(\Delta p(t, :))$
- 7: $ID[t] \leftarrow index$
- 8: **end for**
- 9: **return** ID

q_k^l represents the probability of association between the j^{th} measurement and $\mu_{k|k-1}^l$ through a normal distribution

$$q_k^l \left(\mathbf{z}_k^j \right) = \mathcal{N} \left(\mathbf{z}_k^j; \mathbf{H}_k \boldsymbol{\mu}_{k|k-1}^l, \mathbf{H}_k \mathbf{P}_{k|k-1}^l \mathbf{H}_k^\top + \mathbf{R}_k \right). \quad (13)$$

After the weight and mean have been updated, we prune possible associations to keep computation manageable. To prune effectively, we discard components with weights less than the truncation threshold and by merging components with Mahalanobis distance less than a given merging threshold. The two thresholds are empirically determined to effectively discard false positives while robustly merging close associations during occlusion. Such pruning method yields only the agents with corresponding highest weights w_k^l , and the total remaining number of tracked agents J_k is updated for the next iteration. Finally, the updated state of each track can be computed as a weighted sum of the associated mean as

$$\mathbf{x}_k = \frac{1}{\tilde{w}_k} \sum_{l=1}^{J_k} w_k^l \boldsymbol{\mu}_k^l, \quad w_k^l \leftarrow \max_j w_k^{j,l} \quad \forall j \quad (14)$$

where \tilde{w}_k is the sum of the corresponding weights.

C. Perception Consensus

Our framework does not utilize an external localization system. The multi-agent tracker handles association only in the local frame of each agent. Consequently, a challenge arises to have a global consensus to correctly tag uniqueness and identical tracked agents IDs among all observing drones. Without adopting a strong assumption of knowing the robots' global pose at all times as in [29], we adopt a simple but efficient perception consensus module relying on minimum communication across the swarm. Each agent utilizes its own on-board localization system (e.g., VIO in Section IV), drone detection, and filter to solve the relative tracking problem. A summary of the consensus method is shown in Algorithm 1. It is assumed without loss of generality that each drone also has a unique ID assigned and is aware of its initial position in a global reference frame. The initial alignment between each local robot frame where the on-board localization is defined and the global frame can be computed at take-off using specific objects' visual features. Subsequently, upon startup, the robots are within communication distance and field of view. Once initial tracking on each agent starts with

one of the tracking filters, a one-time broadcast of the initial position of each drone and corresponding ID is performed. Once shared, the 3D position of each drone is re-projected on the image plane, and the re-projection error for all possible tracked agents is computed. Each track ID in the filter of each observing drone will correspond to the drone ID with minimum re-projection error. This procedure can be repeated to restore the consensus if tracking is re-initialized.

III. MULTI-AGENT TRACKING COMPARATIVE ANALYSIS

We benchmark the performance of the Kalman, JPDAF and PHD multi-agent tracking algorithms. We discuss the trade-offs in terms of accuracy and computation complexity under varying operating conditions and number of agents. We simulate 3 target drones crossing their trajectories in the image of an stationary observer drone. Given the unassociated and unsorted measurements, the filter estimates the 2D position tracking of the 3 drones. In the presented tests, the average maximum speed for the 3 drones is around 1 m/s, but similar results hold for different speeds. Simulations are performed on an Intel® i7 quad-core machine.

A. Accuracy vs. Computational Complexity

In Fig. 2, the average tracking error (difference between tracked estimation and simulation ground truth of the 3 drones) for both the JPDAF and GM-PHD filters is shown. We observe that the error is marginally higher for the GM-PHD comparing to the JPDAF filter. The GM-PHD tracking resulted in Root Mean Square Error (RMSE) of 0.735 pixel while JPDAF tracking resulted in 0.302 pixel. The RMSE of Kalman filter is 3.844 pixels, which is much larger than those of the other two filters, so we did not add the RMSE of Kalman filter in Fig. 2 to better highlight the comparison among the two best filters. In Fig. 3, considering the aforementioned setup, we present the processing time of the three filters as a function of the number of tracked drones in log-scale. The Kalman filter's processing time increases linearly across the number of drones tracking. The JPDAF filter shows an exponential computation increase as the number of tracked drones increases, while the GM-PHD filter shows substantially lower computation complexity compared to the JPDAF filter. These results reflect the corresponding computational complexity of the two approaches, namely $O(N)$ for the Kalman filter, $O(N!)$ for the JPDAF [20], [30], and $O(N^2)$ for the PHD filter [29] with N the number of tracked agents. The computation demand becomes untractable for the JPDAF while the number of tracked agents

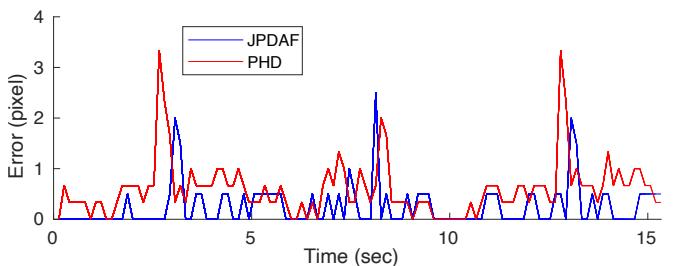


Fig. 2: Average RMSE on tracking 3 drones crossing paths.

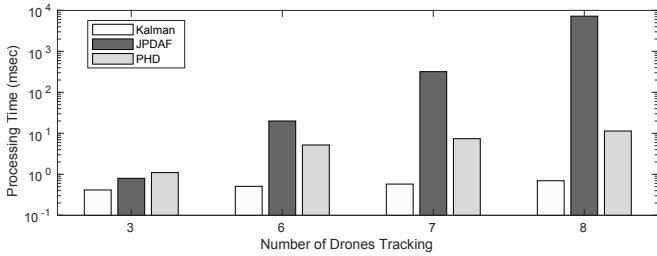


Fig. 3: Computation time varying the number of drones.

in each camera view increases, preventing its deployment for large swarms. Considering 8 drones, the average computation time required at each iteration to resolve the association problem reaches 7.218 s. On the other hand, the PHD filter is significantly faster, employing 0.011 s, thus scaling better to track a large team. However, this comes at the price of a slighter lower accuracy compared to the JPDAF as shown in Fig. 2. Therefore, there is a trade-off between accuracy and computation since the PHD does not compute all associations between measurements and tracked agents as in the JPDAF.

B. Noise Performance Evaluation

We analyze the robustness of the proposed filters with respect to different types and intensities of measurement noise. We vary the number of false positives (number of measurement clutter from the environment, see Fig. 4 left column), injected Gaussian measurement noise (to emulate the noisy camera and IMU readings, see Fig. 4 center column), and false positives (probability of false detections, see Fig. 4 right column) to reflect real-world scenarios. The parameters of both filters have been set accordingly to match the various noise conditions. We show the tracking estimation (Figs. 4(a), 4(c), and 4(b)) with intensities of false positive (1 and 10 cases of clutter), Gaussian noise (25% and 75% of the measurement noise covariance), and false negative (0.97% and 0.80% probability of detection). We also show the tracking errors (Fig. 4(d)). For each case, we vary the intensity and analyze how it affects the tracking performance.

The Kalman filter loses the agent tracking due to poor association as shown in Fig. 4(a) (top row) or even switches tracked agents for large noise value as shown in Fig. 4(a) (bottom row) compared to corresponding plots of other filters in Fig.s 4(b) and 4(c). We can observe that in the evaluation against false positive (see Fig. 4(d) left column), Gaussian noise (see Fig. 4(d) center column), and false negative (see Fig. 4(d) right column) the tracking RMSE is initially higher for the PHD compared to the JPDAF since the summation of Gaussian components affects the track. However, this weighted averaging effect proves to be more robust against high noise where the agent trajectories are crossing paths. The PHD maintains correct association during crossing whereas the JPDAF accrues larger error in intense clutter. For the false positive case of 10 clutters (see Fig. 4(d) left column), the PHD shows 1.81 RMSE that is lower compared to 4.34 of the JPDAF. For Gaussian Noise of 0.75% of measurement noise covariance, PHD also shows 1.74 pixels RMSE which is again lower compared to 2.52 pixels RMSE of the JPDAF. For the false negative case

of 0.8 probability of detection, PHD again shows a lower RMSE of 0.45 compared to 0.62 that of JPDAF. For all other cases with lower noise intensities, we can conclude that the JPDAF shows better performances than the PHD filter. Similar results hold for different speeds. We do not include the result of the Kalman filter in Fig. 4(d) for ease of readability since the scale of the Kalman filter's RMSE is 10 – 100 greater compared to those of the other two filters. The Kalman filter is 477.63 pixels for the false positive with 10 clutters while the JPDAF's RMSE is 4.35 pixels. The high RMSE is due to tracking losses due to poor association in the Kalman filter compared to JPDAF and PHD filters.

IV. EXPERIMENTAL RESULTS

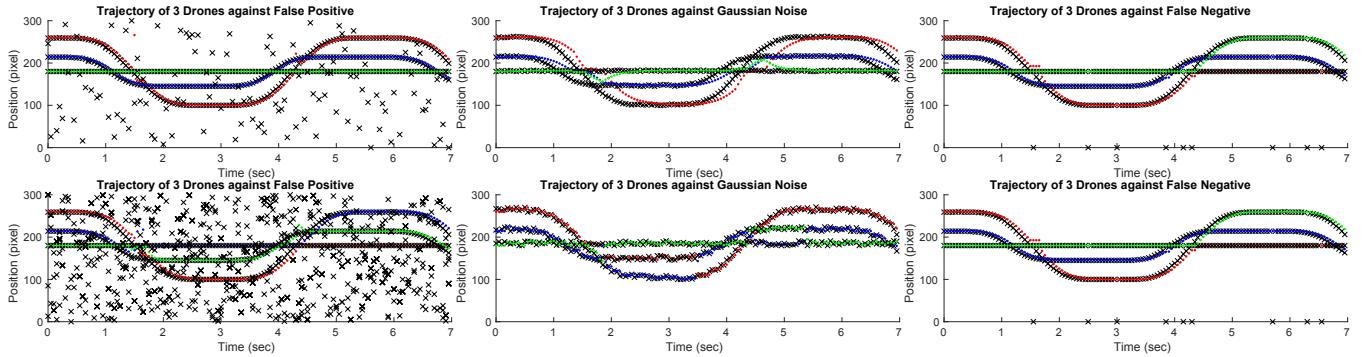
A. System Setup

We report results from experiments with 3 quadrotors conducted in an indoor flying space of $10 \times 6 \times 4$ m³ at the Agile Robotics and Perception Lab (ARPL) lab at New York University. We employ custom small-scale aerial robots equipped with a Qualcomm® Snapdragon™ Flight™ Pro board and on-board VIO, planning, and control based on our previous work [31]. The framework has been developed in ROS. Communication among drones is implemented using a synchronized multi-master network module [32]. Although alternative visual detectors would also work with the proposed tracking strategy, we empirically selected our detector that offered both robustness and inference speed as discussed in the next section. Once the tracking resolves the spatio-temporal association between target drones and the incoming measurements local to each robot, the perception consensus guarantees the uniqueness and identical IDs across all agents.

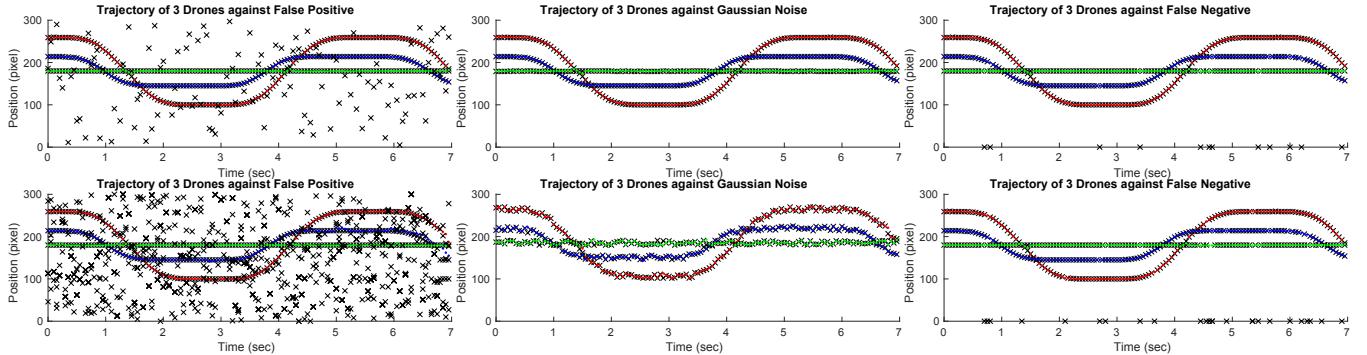
B. Visual Target Detector

We employ a DNN to predict the 2D object centers as well as the regressed 2D bounding boxes from each observing agent's RGB front camera in real-time. Our approach is inherited from CenterNet [33]. By directly regressing the objects' centers, CenterNet provides accurate and robust detection. CenterNet has a better tradeoff between speed and accuracy than YOLOv3 [33]. The architecture is depicted in Fig. 5. Let $I \in \mathcal{R}^{W \times H \times 3}$ be the input image with width W and height H . A ResNet-34 [34] backbone augmented by three up-convolutional networks similar to [33] produces a center keypoint heatmap $\hat{Y} \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times C}$, where R is the output stride (set to 4) and C is the number of classes (set to 1). A local offset $\hat{O} \in \mathcal{R}^{\frac{W}{R} \times \frac{H}{R} \times 2}$ is regressed for each center point to recover the discretization error caused by the output stride. The object size $\hat{S} \in \mathcal{R}^{\frac{W}{R} \times \frac{H}{R} \times 2}$ is also regressed for each center point to obtain the bounding box. We train our model following [33].

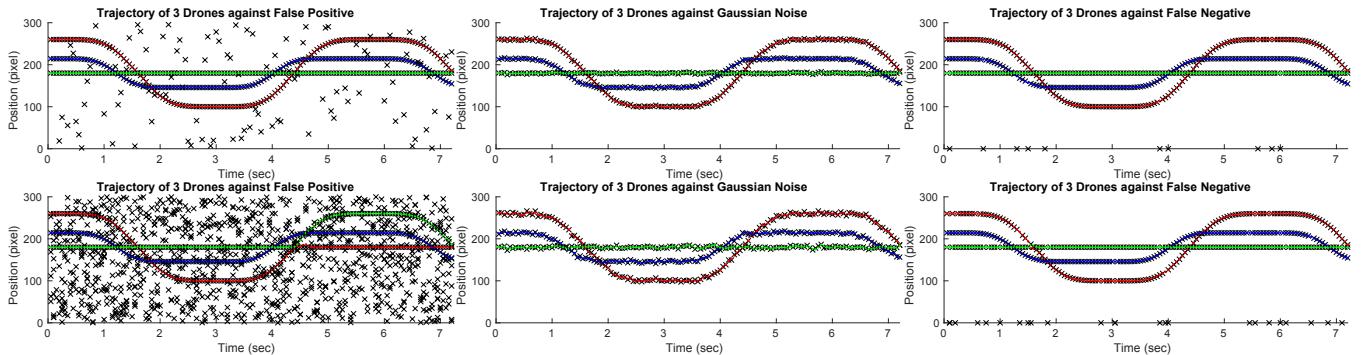
We further adopt Qualcomm® Snapdragon™ Neural Processing Engine (SNPE) to deploy on-board our network. Our solution provides fast and efficient object detection at 7 Hz with on-board GPU. We also noticed that CenterNet-based architectures are easier to deploy on edge devices due to the large number of operators compared to YOLOv3 representing another advantage of this approach.



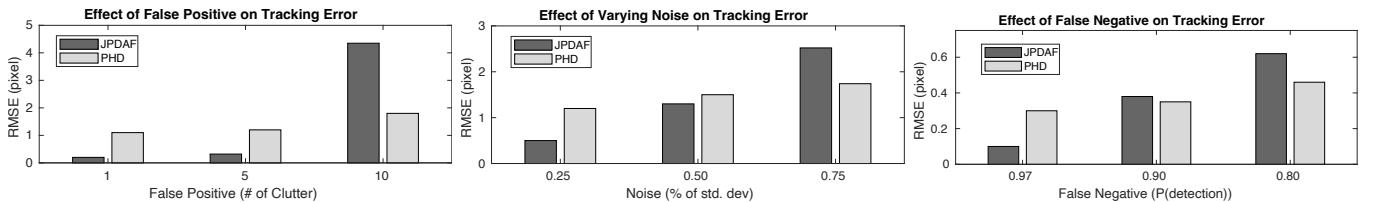
(a) Evaluation for Kalman Filter against false positive, additive noise, and false negative. Increased intensity from top to bottom. The black crosses represent the unassociated raw measurement input. The tracking output of the Kalman filter over time is denoted in blue, red, green for the three targets.



(b) Evaluation for JPDAF filter against false positive, additive noise, and false negative. Increased intensity from top to bottom. The black crosses represent the unassociated raw measurement input. The tracking output of the JPDAF filter over time is denoted in blue, red, green for the three targets.



(c) Evaluation for PHD filter against false positive, additive noise, and false negative. Increased intensity from top to bottom. The black crosses represent the unassociated raw measurement input. The tracking output of the PHD filter over time is denoted in blue, red, green for the three targets.



(d) Comparative analysis of the tracking errors as a function of noise intensities for each type of noise.

Fig. 4: Comparative performance analysis for different noise intensities.

C. Results

To validate our proposed tracking system and compare the performance between Kalman, PHD, and JPDAF filters, we present various experiments where the 3 drones move in each other's field of view. To further test the robustness of the proposed approaches to occlusions, we conduct several experiments where multiple drones are crossing their trajectories in the image of an observing agent. Our filters are

able to estimate the full state of the vehicles at the IMU rate of 100 Hz rather than waiting for a new detection which operates at 7 Hz. This allows to speed up the inference by 14 times and mitigates the problem of incorrect measurement association due to the slow 7 Hz detection rate. Several experiments are also presented in the multimedia material.

In the first experiment, 3 drones move toward the same direction and do not cross trajectories while staying in each

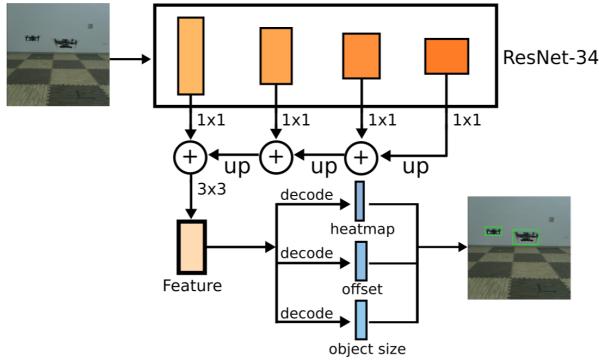


Fig. 5: The proposed DNN architecture.

TABLE I: Filters' accuracy and average computation time.

RMSE↓ [pixels]	drone 1	drone 2	drone 3	Computation Time↓ (ms)
Kalman	15.88	18.52	14.99	4.03
JPDAF	8.94	7.42	9.31	7.83
PHD	12.27	10.48	13.07	8.24

other's field of view. The tracking across the 3 drones' viewpoints is shown in Fig. 6. The maximum relative speed between each target and the observing drone is 0.63 m/s. The CPU usage is 7%. The raw images and all 3 drones' position provided by Vicon are recorded for evaluation. The ground truth position of each drone for each image is obtained by projecting each drone's position on the image. The filter predicts the location of the observed drones while assigning globally consistent and unique IDs. In Table I, we report the RMSE between the ground-truth and estimated positions of the drones in all images for the Kalman, JPDAF, and PHD filters. This experiment has no occlusions or missed detections therefore JPDAF has lower RMSE than that of PHD, which is consistent with the simulation results in Fig. 4(d) for low-medium noise settings. We also report the average computation time in Table I. These are consistent with simulation results in Section III in Fig. 3 since for less than 3 agents, $N = 2$ in this case, the tracking computation cost of Kalman filter is lower compared to JPDAF which is more computationally efficient than PHD filter.

Our second experiment shows that our system is robust to heavy occlusions by testing a hovering observing drone maintains tracks of two drones crossing their trajectories. This setting is challenging since the detection measurements are very close to each other or completely missed during the crossing. This can produce wrong association as shown in Fig. 7. We show that our filters consistently track the drones and maintain correct association for these cases except the Kalman filter. Consistent with what we observed in simulation results, the Kalman filter often incorrectly associates tracks among the agents, resulting in incomparably

TABLE II: Discussion of three filters.

Metric↑	Kalman	JPDAF	PHD
Computation	++++	+	++
Accuracy	+	++++	+++
Robustness	+	+++	++++

high RMSE. In contrast, the averaged RMSE of PHD filter across drones is 6.83 pixels and 8.54 pixels for the JPDAF filter. These results are consistent with the simulation results in Fig. 4(d). We observe that the PHD filter outperforms the JPDAF filter due to the presence of false negative, higher noise and occlusions similar to the simulation results. This experiment has high occlusions since drone trajectories often cross causing missed detections as in Fig. 7.

V. DISCUSSION

Results in Sections III and IV provide several insights on the filter's choice, according to three metrics, summarized in Table II. From a computational perspective, the Kalman filter is the best solution. However, it struggles to be accurate and robust in presence of noise. Conversely, JPDAF and PHD are natural solutions for multi-target tracking providing increased robustness and accuracy. However, the use of the JPDAF is suggested for a small number of tracking agents ($N < 3$). For larger teams, the computation becomes not manageable especially on SWaP constrained robots, therefore the PHD becomes the preferred solution in these cases.

VI. CONCLUSION

In this work, we presented several vision-based decentralized Bayesian filtering strategies for detection and tracking of multiple aerial vehicles and the corresponding perception consensus. We analyzed their trade-offs and infer the most suitable design based on the computation resources and scenarios. Finally, we showed how to transition this perception and inference pipeline onto SWaP constrained robots. The results show the effectiveness and robustness in challenging scenarios, including occlusions and missed detections with the potential to scale to swarms of robots.

Future works will investigate how to perform closed-loop formation control leveraging the tracking information. We envision extending the perception consensus to the absence of communication and incorporating this approach in a multi-modal framework, which can robustify the tracking and contribute to accurate multi-robot localization.

REFERENCES

- [1] S. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A Survey on Aerial Swarm Robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.
- [2] M. Coppola, K. N. McGuire, C. De Wagter, and G. C. H. E. de Croon, "A survey on swarming with micro air vehicles: Fundamental challenges and constraints," *Frontiers in Robotics and AI*, vol. 7, p. 18, 2020.
- [3] F. Schiano and P. Robuffo Giordano, "Bearing rigidity maintenance for formations of quadrotor UAVs," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 1467–1474.
- [4] F. Schiano and R. Tron, "The dynamic bearing observability matrix: Nonlinear observability and estimation for multi-agent systems," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 3669–3676.
- [5] D. Zelazo, A. Franchi, H. H. Bülfhoff, and P. Robuffo Giordano, "Decentralized rigidity maintenance control with range measurements for multi-robot systems," *The International Journal of Robotics Research*, vol. 34, no. 1, pp. 105–128, 2015.
- [6] G. Loianno, Y. Mulgaonkar, C. Brunner, D. Ahuja, A. Ramanandan, M. Chari, S. Diaz, and V. Kumar, "A swarm of flying smartphones," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 1681–1688.



(a) drone 1 tracking viewpoint.

(b) drone 2 tracking viewpoint.

(c) drone 3 tracking viewpoint.

Fig. 6: Tracking performance on 3 drones with current tracked agents (green) and measurements (red).



Fig. 7: Tracking against occlusions. The two rows illustrate the time evolution of two sample challenging drone crossing scenarios with detections (red) and tracker prediction (green). Results show that our tracker is also robust to missed detections.

- [7] A. Weinstein, A. Cho, G. Loianno, and V. Kumar, "Visual inertial odometry swarm: An autonomous swarm of vision-based quadrotors," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1801–1807, 2018.
- [8] R. Tron, J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, "A distributed optimization framework for localization and formation control," *IEEE Control Systems Magazine*, vol. 36, no. 4, pp. 22–44, 2016.
- [9] D. Dias, R. Ventura, P. Lima, and A. Martinoli, "On-board vision-based 3d relative localization system for multiple quadrotors," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1181–1187.
- [10] V. Walter, N. Staub, A. Franchi, and M. Saska, "Uvdar system for visual relative localization with application to leader–follower formations of multirotor uavs," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2637–2644, 2019.
- [11] A. Ahmad, V. Walter, P. Petracek, M. Petrilik, T. Baca, D. Zaitik, and M. Saska, "Autonomous aerial swarming in gnss-denied environments with high obstacle density," in *IEEE/RSJ International Conference on Robotics and Automation*, 2021, pp. 570–576.
- [12] Y. Tang, Y. Hu, J. Cui, F. Liao, M. Lao, F. Lin, and R. S. H. Teo, "Vision-aided multi-uav autonomous flocking in gps-denied environment," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 616–626, 2019.
- [13] D. Thakur, Y. Tao, R. Li, A. Zhou, A. Kushleyev, and V. Kumar, "Swarm of inexpensive heterogeneous micro aerial vehicles," in *Experimental Robotics*, B. Siciliano, C. Laschi, and O. Khatib, Eds. Cham: Springer International Publishing, 2021, pp. 413–423.
- [14] K. Guo, Z. Qiu, W. Meng, L. Xie, and R. Teo, "Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments," *International Journal of Micro Air Vehicles*, vol. 9, no. 3, pp. 169–186, 2017.
- [15] F. J. Perez-Grau, F. Caballero, L. Merino, and A. Viguria, "Multi-modal mapping and localization of unmanned aerial robots based on ultra-wideband and rgb-d sensing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 3495–3502.
- [16] H. Xu, L. Wang, Y. Zhang, K. Qiu, and S. Shen, "Decentralized visual-inertial-uwb fusion for relative state estimation of aerial swarm," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 8776–8782.
- [17] K. R. Sapkota, S. Roelofsen, A. Rozantsev, V. Lepetit, D. Gillet, P. Fua, and A. Martinoli, "Vision-based unmanned aerial vehicle detection and tracking for sense and avoid systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 1556–1561.
- [18] T. Nguyen, K. Mohta, C. J. Taylor, and V. Kumar, "Vision-based Multi-MAV Localization with Anonymous Relative Measurements Using Coupled Probabilistic Data Association Filter," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 3349–3355.
- [19] F. Schilling, F. Schiano, and D. Floreano, "Vision-based drone flocking in outdoor environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2954–2961, 2021.
- [20] M. Pavliv, F. Schiano, C. Reardon, D. Floreano, and G. Loianno, "Tracking and relative localization of drone swarms with a vision-based headset," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1455–1462, 2021.
- [21] X. Tang, X. Chen, M. McDonald, R. Mahler, R. Tharmarasa, and T. Kirubarajan, "A multiple-detection probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 63, no. 8, pp. 2007–2019, 2015.
- [22] J. Smith, F. Particke, M. Hiller, and J. Thielecke, "Systematic analysis of the pmbm, phd, jpd and gnn multi-target tracking filters," in *22th International Conference on Information Fusion*, 2019, pp. 1–8.
- [23] E. Montijano, E. Cristofalo, D. Zhou, M. Schwager, and C. Sagües, "Vision-Based Distributed Formation Control Without an External Positioning System," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 339–351, 2016.
- [24] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual slam," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 2466–2473.
- [25] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.
- [26] R. Ge and G. Loianno, "Vipose: Real-time visual-inertial 6d object pose tracking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 4597–4603.
- [27] F. Cordella, F. Di Corato, G. Loianno, B. Siciliano, and L. Zollo, "Robust pose estimation algorithm for wrist motion tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3746–3751.
- [28] N. Pham, "Tracking of multiple objects using the phd filter," Ph.D. dissertation, Ph. D. dissertation, Department of Electrical and Computer Engineering, 2007.
- [29] P. M. Dames, "Distributed multi-target search and tracking using the PHD filter," *Autonomous Robots*, vol. 44, no. 3-4, pp. 673–689, 2020.
- [30] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 1980, pp. 807–812.
- [31] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, April 2017.
- [32] J. Sergi Hernandez and C. Fernando Herrero, "Multi-master ROS Systems Technical Report," *Institut de Robotica i Informatica Industrial*, 2015.
- [33] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.
- [34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.