drones

MDPI

Article

Distributed Localization for UAV–UGV Cooperative Systems Using Information Consensus Filter

Buqing Ou [1], Feixiang Liu [2] and Guanchong Niu [2,*]

[1] School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510640, China; 202130132419@mail.scut.edu.cn
[2] School of Telecommunications Engineering, Xidian University, Xi'an 710071, China; fx_liu@stu.xidian.edu.cn
* Correspondence: niuguanchong@xidian.edu.cn

Abstract: In the evolving landscape of autonomous systems, the integration of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) has emerged as a promising solution for improving the localization accuracy and operational efficiency for diverse applications. This study introduces an Information Consensus Filter (ICF)-based decentralized control system for UAVs, incorporating the Control Barrier Function–Control Lyapunov Function (CBF–CLF) strategy aimed at enhancing operational safety and efficiency. At the core of our approach lies an ICF-based decentralized control algorithm that allows UAVs to autonomously adjust their flight controls in real time based on inter-UAV communication. This facilitates cohesive movement operation, significantly improving the system resilience and adaptability. Meanwhile, the UAV is equipped with a visual recognition system designed for tracking and locating the UGV. According to the experiments proposed in the paper, the precision of this visual recognition system correlates significantly with the operational distance. The proposed CBF–CLF strategy dynamically adjusts the control inputs to maintain safe distances between the UAV and UGV, thereby enhancing the accuracy of the visual system. The effectiveness and robustness of the proposed system are demonstrated through extensive simulations and experiments, highlighting its potential for widespread application in UAV operational domains.

Keywords: unmanned aerial vehicles; unmanned ground vehicles; Control Lyapunov Function; control barrier function

check for updates

Citation: Ou, B.; Liu, F.; Niu, G. Distributed Localization for UAV–UGV Cooperative Systems Using Information Consensus Filter. *Drones* **2024**, *8*, 166. https://doi.org/10.3390/drones8040166

Academic Editors: Xu Fang, Chao Deng, Shankar A. Deka, Jitao Li and Heling Yuan

Received: 20 March 2024
Revised: 15 April 2024
Accepted: 19 April 2024
Published: 21 April 2024

1. Introduction

In recent years, Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) have captured increasing attention and interest. These vehicles have shown significant potential in various domains, such as search and rescue, surveillance, and environmental monitoring [1]. UAVs and UGVs are typically equipped with a variety of sensors, cameras, and other devices, enabling them to perform tasks including conducting search and rescue operations at disaster sites, monitoring potential danger zones, and environmental surveillance [2]. Their unmanned nature allows them to execute hazardous tasks, thus avoiding risks to human lives [3].

When these UAVs and UGVs form a cooperative system, they complement each other's capabilities, leveraging their respective strengths to accomplish more complex missions [4]. For instance, UAVs can provide high-altitude perspectives and rapid response capabilities [5], while UGVs can offer a stable platform and longer duration of sustained operations [6]. Their collaboration enhances the efficiency of the entire system, improving the effectiveness and accuracy of the work [7].

However, operating these vehicles in a coordinated manner proves to be a challenging task, especially in dynamic and uncertain environments [8]. Localizing UAVs becomes a specific challenge when they operate in tandem with other UAVs or UGVs, particularly in situations where Global Positioning System (GPS) signals are unavailable [9]. This

urgent need for localization has led to the emergence of pioneering studies aimed at addressing this challenge [10]. Moreover, an algorithm is proposed to localize the UAVs using satellite images in [11]. The objective of these studies is to devise inventive solutions that empower UAVs to ascertain their position with precision, even in environments devoid of GPS signals.

Accurate localization information is essential for collaborative operations involving multiple unmanned vehicles. Hence, advancements in sensor technology have played an important role in enhancing the localization capabilities of UAVs and UGVs [12]. For instance, the integration of visual odometry, which relies on the analysis of camera images to estimate the motion of a vehicle, has emerged as a promising approach in environments where traditional GPS-based navigation is ineffective. Additionally, the use of Light Detection and Ranging (LiDAR) and radar technologies for mapping and navigation in complex terrains has gained traction [13]. These technologies enable UAVs and UGVs to create high-resolution maps of their surroundings, facilitating more precise positioning and maneuvering. Nevertheless, sensor technology may face limitations in precision, leading to less accurate data acquisition and consequently impacting the precision of localization information. Moreover, some sensors may exhibit sensitivity to environmental conditions such as changes in weather and variations in lighting, potentially compromising the sensor performance and, in turn, affecting the localization accuracy [14].

Simultaneously, machine learning algorithms, specifically within deep learning, have significantly advanced the localization capabilities of UAVs and UGVs. In [15], a vision-based algorithm is introduced to localize the targets using the UAV–UGV cooperative systems. Through the utilization of neural networks, these vehicles are now capable of processing extensive sensor data in real time, enabling them to make more precise and adaptable decisions [16]. Such capability proves critical in scenarios characterized by rapidly changing environmental conditions [17]. Conversely, deep learning algorithms often necessitate extensive datasets for training to achieve precise models [18]. Acquiring substantial training data in UAVs and UGVs might be constrained, particularly in intricate or perilous environments. Deep learning models might exhibit sensitivity to alterations in the environment, necessitating retraining or recalibration when the environmental conditions fluctuate, potentially impacting the accuracy of the localization information.

Furthermore, the integration of UAVs and UGVs into cooperative systems also offers prospects for leveraging swarm intelligence [19]. Through coordinated operations, these vehicles can exchange localization data and insights, enhancing the accuracy and efficiency of their endeavors. This swarm-based approach not only bolsters the capabilities of individual vehicles but also extends their operational scope and resilience in demanding environments [20]. In [21], a coalition formation game approach is proposed to localize the UAV swarm in a GNSS-denied environment.

Moreover, the establishment of standardized communication protocols and interoperable software frameworks is imperative for the smooth integration of UAVs and UGVs into a collaborative system [22]. These frameworks guarantee dependable and efficient data exchange and coordination among diverse vehicle types, thereby contributing their overall performance as a collective entity.

To cope with the aforementioned challenges, the expanding potential applications of UAVs and UGVs within cooperative systems signify a significant advancement in autonomous technology. Their versatility across various sectors, from agricultural monitoring [23] to urban planning [24], demonstrates their adaptability to diverse environments and tasks. Particularly notable is their ability to operate effectively in GPS-denied environments, overcoming the existing limitations and unlocking new opportunities for these technologies. This adaptability is crucial in addressing the challenges posed by inaccessible or compromised GPS signals, ensuring uninterrupted operations and broadening the application scope for UAVs and UGVs. Consequently, their capability to autonomously navigate and perform tasks in such conditions represents an essential step forward in the evolution of autonomous vehicle technology.

Recently, numerous solutions for UAV–UGV collaboration have emerged. UAVs and UGVs have garnered attention for their potential in high-risk missions, necessitating an effective cooperative control framework to optimize mission completion and resource utilization, particularly in applications like wildfire fighting [25]. However, their adaptability to dynamic environments may pose constraints, and relying on a central mobile mission controller for decisionmaking and planning could elevate the risk of system failure due to a single point of failure. In hazardous industrial settings, a decentralized scheme employs a drone to lead ground mobile robots transporting objects, with one robot navigating obstacles using drone-provided waypoints and a human operator while the others maintain distance and bearing through predictive vision-based tracking [26]. However, this scheme places high demands on the reliability and precision of both the drone and human operator, potentially leading to a lack of robustness. Failures or errors regarding either the drone or the operator could adversely affect the overall performance of the system. Reference [27] presents a cooperative UAV–UGV team for aerial tasks, proposing a control framework validated experimentally for efficient operation and reduced energy use. While UAVs and UGVs can assist each other, ensuring they do not cause damage or pose danger during task execution remains a challenge in real-world environments.

Distributed algorithms are computational procedures designed for systems consisting of multiple computing nodes [28]. These algorithms distribute tasks among the nodes, enabling them to collaborate and communicate in achieving a common objective. Each node in a distributed system possesses its own computational resources and local state [29]. Distributed algorithms play a critical role in the coordination and operation of UAVs and UGVs. In [30], a distributed solution mechanism is employed to determine the information-maximizing trajectories subject to the constraints of individual vehicle and sensor sub-systems. Additionally, ref. [31] demonstrates a considerable advantage of distributed UGVs over the static placement of control stations. These distributed algorithms facilitate communication and collaboration among the UAVs and UGV, allowing them to work together effectively to accomplish tasks.

The Information Consensus Filter (ICF) is a distributed algorithm used in multi-agent systems for estimating a common state [32]. It iteratively updates each agent's estimate based on the local observations and received information from neighboring agents, using a consensus matrix to determine the weights of these updates. Through iterative consensus iterations, the ICF enables the agents to converge towards a consistent estimate of the common state, making it valuable for distributed estimation and control in dynamic environments. The ICF in UAV and UGV systems promotes information exchange and maintains state coherence across various Unmanned Aerial Vehicles and Unmanned Ground Vehicles. By fostering communication and collaboration among individual UAVs and UGVs, the ICF ensures a unified estimation of the shared states, thereby optimizing the task performance and resource utilization. Its implementation bolsters the system coordination and resilience, allowing for adept navigation in intricate environments and expanded utility across diverse applications. In [33], a distributed information filter algorithm is developed for each UGV to locally estimate the position and velocity of the quadrotor using its information and information from neighboring UGVs. Despite the pioneering advantages, the aforementioned research overlooks the integration of localization and control in unmanned vehicles.

1.1. Motivations

Improving the robustness and reliability of cooperative UAV–UGV systems is driven by the need to overcome challenges. This involves lessening the burdens on individual drones and human operators, ensuring adaptability to changing environments, and decreasing dependence on centralized control. The goal is to enhance the overall performance and safety of these systems in real-world scenarios. In response to the complexities of coordinating vehicle operations and guaranteeing precise localization data, we devised and executed a cost-effective vision-centered collaborative framework. While fully considering

the superiority of distributed algorithms such as the ICF, our framework draws inspiration from the integration of UAVs and UGVs into cooperative systems and also incorporates the establishment of standardized communication protocols. It is engineered to dynamically adapt to changing environmental conditions, prioritize vehicle safety, and establish a reliable and efficient communication infrastructure. Additionally, it is engineered to accommodate multiple vehicles and facilitate seamless collaboration among them.

1.2. Contributions

We have developed and implemented a system that incorporates an advanced decentralized control algorithm and employs a Control Barrier Function–Control Lyapunov Function (CBF–CLF) strategy for UAVs, aimed at enhancing safety and efficiency. The efficacy of our system is validated through simulations of real-world scenarios.

In summary, the contributions are elaborated as follows:

- **Distributed Algorithm Design:** Our system integrates an advanced decentralized control algorithm, allowing UAVs to autonomously adapt control inputs in real time by leveraging data from the Information Consensus Filter (ICF). This decentralized strategy enables each UAV to function independently yet maintain cohesion within a larger fleet, bolstering system resilience and adaptability.
- **Control Algorithm Design:** At the core of our system is the Control Barrier Function–Control Lyapunov Function (CBF–CLF) strategy. This innovative control scheme emphasizes the safety of UAVs, which is essential for ensuring operational integrity. By enforcing safe separation distances between UAVs and obstacles, the CBF–CLF strategy minimizes collision risks. The continuous adjustment of control inputs, informed by real-time assessments of the vehicle's state and surroundings, ensures effective implementation.
- **Real-World Implementation:** We evaluated the effectiveness and robustness of our system through extensive simulations in various real-world scenarios. This practical testing confirms the readiness of our system for real-world applications and sets the stage for its deployment in diverse UAV operational domains.

In comparison to the existing research methods, our system addresses several key limitations. Despite the high demands placed on the reliability and precision of both the drone and human operator, which could potentially lead to a lack of robustness, our system mitigates this risk through the incorporation of an advanced decentralized control algorithm. This algorithm enables UAVs to independently adjust the control inputs in real time via the Information Consensus Filter (ICF), reducing the dependency on precise human intervention and enhancing the overall system resilience. Additionally, while their adaptability to dynamic environments may pose constraints, our system's decentralized approach allows for greater flexibility in responding to changing conditions. Furthermore, the reliance on a central mobile mission controller for decisionmaking and planning, which could elevate the risk of system failure due to a single point of failure, is minimized in our system by distributing the decisionmaking processes across the UAV fleet. Finally, while UAVs and UGVs can assist each other, ensuring that they do not cause damage or pose danger during task execution remains a challenge in real-world environments. However, our system addresses this challenge through the implementation of the Control Barrier Function–Control Lyapunov Function (CBF–CLF) strategy, which prioritizes UAV safety by enforcing safe separation distances from obstacles, thereby reducing collision risks. Through extensive simulation-based evaluations across various real-world scenarios, our system's effectiveness and robustness were confirmed, affirming its readiness for diverse UAV operational domains.

The rest of this paper is organized as follows. Section 2.1 describes the preliminary knowledge of the distributed information for the cooperative systems and Control Lyapunov/Barrier Function. Afterwards, Section 2.2 provides the details regarding the proposed ICF-based Distributed UAV–UGV Cooperative System, including the system overview, dynamics, ICF for UAVs, and CBF–CLF for UAV–UGV Coopeative System. After that, extensive experimental results are presented in Section 3. Finally, the conclusion is provided in Section 4.

2. Materials and Methods

2.1. Preliminary Knowledge

2.1.1. Information-Weighted Consensus Filter

In the field of distributed systems, the ICF algorithm has emerged as a powerful algorithm for achieving consensus among multiple agents, even with limited communication and information exchange. It enables distributed entities to collaboratively estimate a common state or variable, utilizing local measurements and restricted information sharing.

Envision a distributed system consisting of N agents, indicated as $A_1, A_2, \ldots, A_N$. Every agent A_i conducts its own local measurement and assesses the overarching state or variable. The ICF's objective is to foster a consensus on this global state estimation through a process of iterative updates and reciprocal information sharing among the agents. The system's linear dynamic system is characterized as follows:

$$\boldsymbol{x}(t+1) = \phi \boldsymbol{x}(t) + \gamma(t), \tag{1}$$

where $\boldsymbol{x}$ is the state vector, ϕ is the state transfer matrix, $\gamma \in (0, Q)$ is the process noise, and $\boldsymbol{Q}$ is the process covariance.

At time t, node A_i possesses the following inputs: prior state estimation $\hat{\boldsymbol{x}}i^-(t)$, prior information matrix $\boldsymbol{J}i^-(t)$, observation matrix $\boldsymbol{H}_i$, consensus rate parameter ϵ, and total consensus iterations K. Upon acquiring the measurement value $\boldsymbol{z}_i$ and the measurement covariance matrix $\boldsymbol{R}_i^{-1}$, the initialization of the consistency information parameters will be conducted using the following equation:

$$\boldsymbol{V}_i^0 \leftarrow \frac{1}{N}\boldsymbol{J}_i^-(t) + \boldsymbol{H}_i^T \boldsymbol{R}_i^{-1} \boldsymbol{H}_i, \tag{2}$$

$$\boldsymbol{v}_i^0 \leftarrow \frac{1}{N}\boldsymbol{J}_i^-(t)\hat{\boldsymbol{x}}_i^-(t) + \boldsymbol{H}_i^T \boldsymbol{R}_i^{-1} \boldsymbol{z}_i. \tag{3}$$

Here, $\boldsymbol{V}_i^0$ and $\boldsymbol{v}_i^0$ represent the initial information matrix and vector, respectively. Subsequently, $\boldsymbol{V}_i^{k-1}$ and $\boldsymbol{v}_i^{k-1}$ are updated using the average consistency algorithm over K iterations. During each iteration, node $A_{i'}$ transmits $\boldsymbol{V}_i^{k-1}$ and $\boldsymbol{v}_i^{k-1}$ to all neighboring nodes $A_{i'}$, while simultaneously receives $\boldsymbol{V}_{i'}^{k-1}$ and $\boldsymbol{v}_{i'}^{k-1}$ sent from all neighboring nodes. The information is updated according to the following equation:

$$\boldsymbol{V}_i^\kappa \leftarrow \boldsymbol{V}_i^{\kappa-1} + \epsilon \sum_{i' \in \mathcal{N}_i} (\boldsymbol{V}_{i'}^{\kappa-1} - \boldsymbol{V}_i^{\kappa-1}), \tag{4}$$

$$\boldsymbol{v}_i^\kappa \leftarrow \boldsymbol{v}_i^{\kappa-1} + \epsilon \sum_{i' \in \mathcal{N}_i} (\boldsymbol{v}_{i'}^{\kappa-1} - \boldsymbol{v}_i^{\kappa-1}). \tag{5}$$

Next, the posterior state estimate and information matrix for time t can be computed using the information vector and information matrix after iterative updating. The equations are as follows:

$$\hat{\boldsymbol{x}}_i^+(t) \leftarrow (\boldsymbol{V}_i^K)^{-1} \boldsymbol{v}_i^K, \tag{6}$$

$$\boldsymbol{J}_i^+(t) \leftarrow N\boldsymbol{V}_i^K. \tag{7}$$

Finally, state information for time $t+1$ can be predicted using following equation:

$$\hat{\boldsymbol{x}}_i^-(t+1) \leftarrow \boldsymbol{\phi}\hat{\boldsymbol{x}}_i^+(t), \tag{8}$$

$$\boldsymbol{J}_i^-(t+1) \leftarrow (\boldsymbol{\phi}(\boldsymbol{J}_i^+(t))^{-1}\boldsymbol{\phi}^T + \boldsymbol{Q})^{-1}. \tag{9}$$

The ICF refines each agent's estimates by incorporating the weighted average of estimates from neighboring agents. Through continuous information exchange and iterative updates, the agents progressively converge to reach a consensus on the estimated global state or variable.

In the subsequent sections, we will delve into the practical applications of the ICF in distributed systems.

2.1.2. Control Lyapunov/Barrier Function

For a continuous-time affine control system $\dot{\boldsymbol{x}} = F(t, \boldsymbol{x}, u)$, where $\boldsymbol{x} \in \mathbb{R}^n$ represents the state vector and $u \in \mathbb{R}^m$ denotes the control input, if F is Lipschitz continuous with respect to $\boldsymbol{x}$ and u, and time t is piecewise continuous while u is also piecewise continuous at time t, then, given the initial conditions, the trajectory exists and remains unique. Given a dynamical system where $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, $g : \mathbb{R}^n \longrightarrow \mathbb{R}^{n\times m}$ and $\boldsymbol{x}$ is Lipschitz continuous

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})u. \tag{10}$$

If there exists a constant C such that the continuous differential function $V(\boldsymbol{x})$ satisfies the following conditions, then $V(\boldsymbol{x})$ is considered a Control Lyapunov Function with respect to $\boldsymbol{x}$:

1. $\Omega_c := \{ \boldsymbol{x} \in \mathbb{R}^n V(\boldsymbol{x}) :\leq C \}$
2. $V(\boldsymbol{x}) > 0, \forall s \in \mathbb{R}^n \backslash \{ \boldsymbol{x}_e \}, V(\boldsymbol{x_e}) = 0$
3. $\inf_{u\in U} \dot{V}(\boldsymbol{x}, u) < 0, \forall \boldsymbol{x} \in \Omega_c, \backslash \{ \boldsymbol{x}_e \}$.

If $V(\boldsymbol{x}) : \mathbb{R}^n \longrightarrow \mathbb{R}$ is continuously differentiable, positive definite, and radially unbounded, then $V(\boldsymbol{x})$ can be proven to be an Exponentially Stabilizing Control Lyapunov Function (ESCLF) using the equation, where λ represents the upper bound on the decay rate of the Lyapunov function:

$$\exists u \in U, \lambda > 0, \dot{V}(\boldsymbol{x}, u) + \lambda V(\boldsymbol{x}) \leq 0. \tag{11}$$

When $\boldsymbol{x}$ belongs to the following set and $h(\dot{\boldsymbol{x}}) \geq 0$, then $h(\boldsymbol{x})$ is considered to satisfy the conditions of Control Barrier Function:

1. $\mathcal{C} = \boldsymbol{x} \in D \subset \mathbb{R}^n : h(\boldsymbol{x}) \geq 0$
2. $\partial\mathcal{C} = \boldsymbol{x} \in D \subset \mathbb{R}^n : h(\boldsymbol{x}) = 0$
3. $Int(\mathcal{C}) = \boldsymbol{x} \in D \subset \mathbb{R}^n : h(\boldsymbol{x}) > 0$

In practical systems, a lower bound on the attenuation rate γ is usually specified for the system if $B(\boldsymbol{x}) : \mathbb{R}^n \longrightarrow \mathbb{R}$ is continuously differentiable, and u making $B(\boldsymbol{x})$ satisfy Equation (12) will keep the system safe at all times.

$$\exists u \in U, \gamma > 0, \dot{B}(\boldsymbol{x}, u) + \gamma B(\boldsymbol{x}) \geq 0. \tag{12}$$

CLF ensures stability of the controller, whereas CBF guarantees that the system meets safety constraints. Integrating these two approaches allows for the formulation of a quadratic programming problem.

The CBF–CLF QP function is defined as

$$\begin{aligned} u^* &= \arg \min_{u=[u,\delta]^T} \frac{1}{2} u^T \boldsymbol{H} u + \boldsymbol{F}^T u \\ s.t. &\quad \begin{cases} L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x}) u + \lambda V(\boldsymbol{x}) \leq \delta, \\ L_f B(\boldsymbol{x}) + L_g B(\boldsymbol{x}) u + \gamma B(\boldsymbol{x}) \geq 0, \end{cases} \end{aligned} \tag{13}$$

where $V(\boldsymbol{x})$ is the CLF function, $B(\boldsymbol{x})$ is the CBF function, $\boldsymbol{H}$ is a positive definite matrix, while $\boldsymbol{F} \in \mathbb{R}^{n\times m}$, δ serves as a relaxation factor.

The CBF–CLF QP function offers a computationally efficient method to ensure both safety and stability of the controlled system. By resolving the quadratic program, an

optimal control input can be derived to minimize the cost function while satisfying the safety and input constraints.

In the subsequent sections, we will delve into the theoretical foundations and practical applications of the CBF–CLF QP function in control systems.

As shown in Figure 1, the system configuration comprises two UAVs assigned to track a UGV, with an "Armor" object mounted on the UGV for positioning purposes. This setup allows the UAV's camera to utilize visual algorithms to calculate the UGV's pose. Initial visual experiments indicate that maintaining a specific range between the UAVs and the UGV yields optimal performance. Leveraging this observation, the CBF–CLF equation is developed. Moreover, to improve the precision of the gathered data, the two UAVs exchange information and refine the collected data using the ICF algorithm.

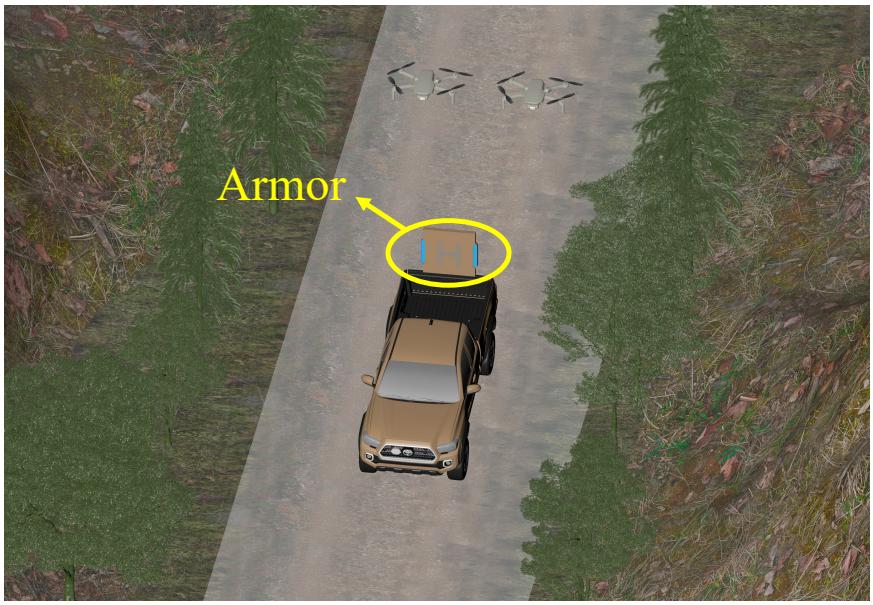

Figure 1. The schematic diagram of UAV–UGV cooperative system.

2.1.3. Vision-Based Localization for UAVs

Image processing is carried out using OpenCV—4.8.1 https://opencv.org/ (accessed on 27 September 2023, developed by Intel, Santa Clara, CA, USA). The camera captures an image, which is then converted to grayscale. The contours of the light bars are detected, and an algorithm is utilized to precisely identify the four vertices of the light bars for localization purposes, along with extracting other pertinent information.

The visual feature, denoted as "Armor" and employed for identification in this experiment, is illustrated in Figure 2. It consists of two light bars mounted parallel on a rectangular plate, both of equal length and width. Upon capturing the contours, the OpenCV boundingRect function is utilized to determine the minimum bounding rectangle encompassing the light bars. In an image, aside from the target identification object, various noise may be present, which can be eliminated by applying filters based on the geometric characteristics of the contours.

Considering variations in the distance and angles between the UAVs and UGV during movement, the mentioned constraints can be suitably relaxed within a specific range. The contours acquired after filtering should precisely depict the target light bars. By mathematical sorting, the highest and lowest points on the contours of the two light bars are discerned, determining the four boundary points of the rectangle. These points are subsequently employed for pose calculation.

Figure 2. Armor (light bars are the main visual features; A, B, C, and D are four points used for solving pose).

In this paper, the PNP algorithm is used to calculate the camera's pose. To enhance the accuracy and stability of the PNP solution, we base our calculations on the following assumption, where l_1 is the Armor plane, l_2 is the groud, l_3 is the camera plane.

Assumption 1. *$|\overrightarrow{AB}|$ is known and $l_1 \parallel l_2 \parallel l_3$, while $\overrightarrow{AB} \overset{\parallel}{=} \overrightarrow{DC}$.*

Assumption 2. *Each UAV can communicate with its neighbors, transferring the relative position of moving target.*

PNP, commonly used in 3D reconstruction and camera pose estimation, follows a typical 3D–2D pose estimation process. It requires real coordinates of N spatial points in a known world coordinate system and their projections onto the image plane. At least three pairs of points are necessary for this process. In this scenario, the 3D points in the world coordinate system and the 2D points in the image coordinate system are known, while the camera pose remains unknown.

In this paper, we employ the Direct Linear Transform (DLT) method for pose calculation, specifically using the solvePNP function in OpenCV. Geometric structure of PNP is shown in Figure 3.

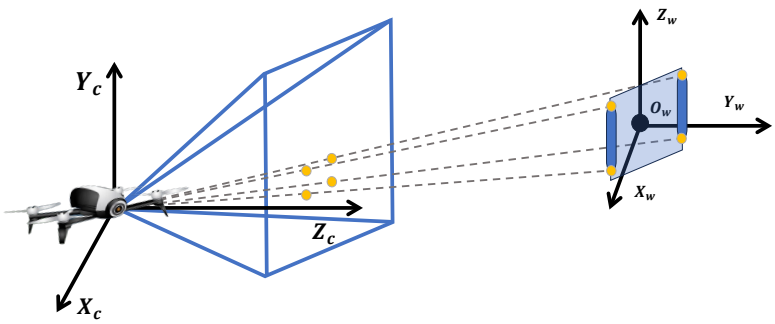

Figure 3. Geometric structure of PNP.

The 3D coordinates of the Armor on the UGV in the world coordinate system can be expressed as

$$\boldsymbol{c} = \left[x_w, y_w, z_w, 1\right]^T. \tag{14}$$

The 2D projected point of Armor in the image coordinate system can be expressed as

$$\boldsymbol{h} = \left[u, v, 1\right]^T. \tag{15}$$

The intrinsic matrix of the camera can be expressed as

$$\boldsymbol{w} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{16}$$

Therefore, the relationship between 2D points in camera image and 3D points in real world can be expressed as

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{c|c} \boldsymbol{R} & \boldsymbol{t} \end{array} \right] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \tag{17}$$

where $\boldsymbol{R}$ and $\boldsymbol{t}$ represent the rotation matrix and translation vector, respectively, describing the Armor pose to be determined. Substituting the four sets of solutions will yield the final pose solution.

2.2. Methodology

2.2.1. Icf-Based Distributed UAV–UGV Cooperative System

UAV–UGV cooperative systems often operate in dynamic and unpredictable environments. Adapting localization algorithms to maintain accuracy in real time poses a considerable challenge. Techniques for restricting UAV within the working range for optimal camera performance while ensuring computational efficiency are needed. Additionally, as the number of UAVs and UGVs in the cooperative system increases, the complexity of distributed localization grows exponentially. Algorithms capable of handling large-scale deployments while maintaining accuracy and robustness are essential. Meanwhile, ensuring the resilience of the localization system to sensor failures and communication disruptions is critical to information processing.

Based on the previous discussion, we have innovatively proposed the ICF-based Distributed UAV–UGV Cooperative System, providing a solution to the above-mentioned problems. The implementation of CBF–CLF equations serves to regulate the distance between the UAVs and UGV, ensuring data accuracy and integrity. By limiting the distance, each sensor operates within its optimal performance range. Moreover, to achieve complete coverage within the UAV's FOV, we increase the number of UAVs and strategically position them. Additionally, we employ the ICF algorithm to facilitate data exchange among UAVs, enabling data fusion to address the issue of naive nodes and enhance state estimation. In essence, the two algorithms in the proposed system run independently, providing a practical dual guarantee of data accuracy from different aspects, optimizing individual camera data computations while ensuring data integrity and robust state estimation. The schematic diagram of ICF-based Distributed UAV–UGV Cooperative System is shown in Figure 4.

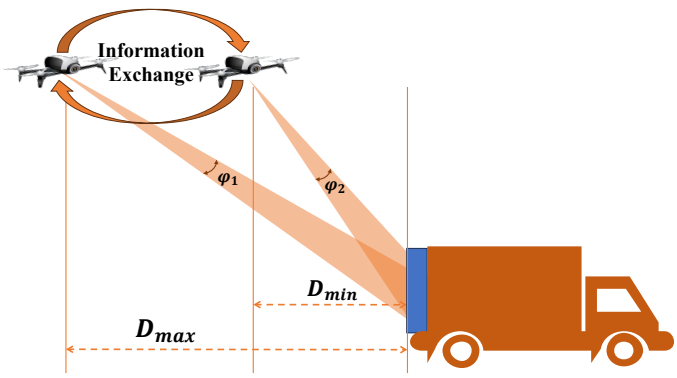

Figure 4. ICF-based Distributed UAV–UGV Cooperative System.

2.2.2. Dynamics

We commence by formulating the system dynamics. Our model assumes linear trajectories for both the UAV and the UGV, with both vehicles consistently maintaining aligned directions.

The velocity of UGV $\boldsymbol{v}_l(t)$ can be calculated using Equation (18), where $\boldsymbol{D}(t)$ is the relative displacement of the UGV, $\boldsymbol{v}_f$ is the velocity of UAV, and ΔT is the time interval between t and $t+1$.

$$\boldsymbol{v}_l(t) = \boldsymbol{v}_f(t) + \frac{\boldsymbol{D}(t) - \boldsymbol{D}(t-1)}{\triangle T}. \tag{18}$$

During the simulation, $\boldsymbol{v}_f(t)$ will be updated at each time t using $\boldsymbol{\alpha}_l(t)$, the acceleration of UAV. It can be updated by

$$\boldsymbol{v}_f(t) = \boldsymbol{v}_f(t+1) + \boldsymbol{\alpha}_f(t) \triangle T. \tag{19}$$

The acceleration of the UAV is represented as

$$\boldsymbol{\alpha}_f(t) = \frac{u(t) - F_r(\boldsymbol{v}_f(t))}{M}. \tag{20}$$

The dynamics of the system can be expressed as

$$\dot{\boldsymbol{x}}(t) = \underbrace{\begin{bmatrix} -F_r(\boldsymbol{v}_f(t))/M \\ \boldsymbol{\alpha}_f(t) \\ \boldsymbol{v}_l(t) - \boldsymbol{v}_f(t) \end{bmatrix}}_{f(\boldsymbol{x}_i)(t)} + \underbrace{\begin{bmatrix} 1/M \\ 0 \\ 0 \end{bmatrix}}_{g(\boldsymbol{x}_i)(t)} u(t). \tag{21}$$

Here, $x = (x_1, x_2, x_3) = (\boldsymbol{v}_f(t), \boldsymbol{v}_l(t), \boldsymbol{D}(t))$, M is the mass of the UAV, and $u(t)$ is the control input of UAV. In this equation, $\boldsymbol{v}_f(t)$ can be regulated by feedback through $u(t)$. The aerodynamic drag values with constants f_0, f_1, and f_2 determined empirically are expressed as

$$F_r(\boldsymbol{v}_f(t)) = f_0 + f_1 \boldsymbol{v}_f(t) + f_2 \boldsymbol{v}_f^2(t). \tag{22}$$

2.2.3. ICF for UAVs

The field of view (FOV) of a camera is typically limited. To ensure comprehensive coverage, it is common practice to deploy two or more UAVs equipped with cameras for tracking and measurement purposes. This arrangement guarantees that the trajectory of the UGV can be fully observed, and the fusion of measurement data from multiple UAVs significantly enhances the accuracy of the measurements.

We consider a continuous linear motion model with constant velocity:

$$\boldsymbol{x}(t+1) = \boldsymbol{M}\boldsymbol{x}(t) + \boldsymbol{\gamma}(t), \tag{23}$$

where $\boldsymbol{M}$ is the state transition matrix, while $\boldsymbol{\gamma}(t) \in N(0, \boldsymbol{Q})$ is Gaussian process noise with positive definite covariance matrix $\boldsymbol{Q}$.

During each time step t during the operation of the ICF, it requires K iterations to achieve consensus with neighboring UAVs regarding the current estimate of the UGV trajectory and the associated information matrix. This process is designed to maintain alignment with the neighboring estimates and pertinent information matrices related to the current UGV trajectory. Through multiple iterations, the ICF endeavors to synchronize and update information, thereby enhancing the overall consensus.

Simultaneously, it is crucial to ensure that the UGV remains continuously within the FOV of the UAVs. Importantly, it suffices for the UGV to be within the FOV of any UAV. Through the exchange of information with neighboring UAVs, the algorithm can partially extend the observation range, thereby maintaining data integrity. This characteristic

represents a significant advantage of the algorithm. The average consensus algorithm is articulated as follows:

$$\boldsymbol{x}_i^{\kappa} \leftarrow \boldsymbol{x}_i^{\kappa-1} + \epsilon \sum_{j \in \mathcal{N}_i} (\boldsymbol{x}_j^{\kappa-1} - \boldsymbol{x}_i^{\kappa-1}). \tag{24}$$

Prediction for the next time step t can be expressed as

$$\hat{\boldsymbol{x}}_i^{-}(t+1) \leftarrow \boldsymbol{M}\hat{\boldsymbol{x}}_i^{+}(t), \tag{25}$$

$$\boldsymbol{G}_i^{-}(t+1) \leftarrow (\boldsymbol{M}(\boldsymbol{G}_i^{+}(t))^{-1}\boldsymbol{M}^T + \boldsymbol{Q})^{-1}. \tag{26}$$

Therefore, we obtain the ICF-based UAV–UGV cooperative system in Algorithm 1.

Algorithm 1: ICF at UAV i relative to UGV r at time step k

Input: Prior state estimate $\hat{\boldsymbol{x}}_i^{-}(t)$, prior information matrix $\boldsymbol{G}_i^{-}(t)$, observation matrix $\boldsymbol{H}_i$, consensus rate parameter ϵ, total consensus iterations K, and process covariance $\boldsymbol{Q}$.

(1) Obtain measurement vector $\boldsymbol{z}_i$ and matrix $\boldsymbol{B}_i$

(2) Compute initial information matrix $\boldsymbol{V}_i^0$ and vector $\boldsymbol{v}_i^0$

$$\boldsymbol{V}_i^0 \leftarrow \frac{1}{N}\boldsymbol{G}_i^{-}(t) + \boldsymbol{H}_i^T\boldsymbol{B}_i^{-1}\boldsymbol{H}_i$$

$$\boldsymbol{v}_i^0 \leftarrow \frac{1}{N}\boldsymbol{G}_i^{-}(t)\hat{\boldsymbol{x}}_i^{-}(t) + \boldsymbol{H}_i^T\boldsymbol{B}_i^{-1}\boldsymbol{z}_i$$

(3) Perform average consensus on $\boldsymbol{V}_i^0$ and $\boldsymbol{v}_i^0$ independently

for $\kappa = 1$ **to** K **do**

(a) Send $\boldsymbol{V}_i^{\kappa-1}$ and $\boldsymbol{v}_i^{\kappa-1}$ to all neighbors $j \in \mathcal{N}_i$

(b) Receive $\boldsymbol{V}_i^{\kappa-1}$ and $\boldsymbol{v}_i^{\kappa-1}$ from all neighbors $j \in \mathcal{N}_i$

(c) Update

$$\boldsymbol{V}_i^{\kappa} \leftarrow \boldsymbol{V}_i^{\kappa-1} + \epsilon \sum_{j \in \mathcal{N}_i} (\boldsymbol{V}_j^{\kappa-1} - \boldsymbol{V}_i^{\kappa-1})$$

$$\boldsymbol{v}_i^{\kappa} \leftarrow \boldsymbol{v}_i^{\kappa-1} + \epsilon \sum_{j \in \mathcal{N}_i} (\boldsymbol{v}_j^{\kappa-1} - \boldsymbol{v}_i^{\kappa-1})$$

end

(4) Compute posteriori state estimate $\hat{\boldsymbol{x}}_i^{+}(t)$ and information matrix $\boldsymbol{G}_i^{+}(t)$ for time t

$$\hat{\boldsymbol{x}}_i^{+}(t) \leftarrow (\boldsymbol{V}_i^K)^{-1}\boldsymbol{v}_i^K$$

$$\boldsymbol{G}_i^{+}(t) \leftarrow N\boldsymbol{V}_i^K$$

(5) Predict for next time step $(t+1)$

$$\hat{\boldsymbol{x}}_i^{-}(t+1) \leftarrow \boldsymbol{M}\hat{\boldsymbol{x}}_i^{+}(t)$$

$$\boldsymbol{J}_i^{-}(t+1) \leftarrow (\boldsymbol{M}(\boldsymbol{G}_i^{+}(t))^{-1}\boldsymbol{M}^T + \boldsymbol{Q})^{-1}$$

Output: State estimate $\hat{\boldsymbol{x}}_i^{+}(t)$ information matrix $\boldsymbol{G}_i^{+}(t)$.

2.2.4. CBF–CLF for UAV–UGV Coopeative System

Based on the system's dynamics model, we propose a soft constraint for the system, aiming to enable the UAV to achieve a desired speed.

$$V(\boldsymbol{x}) = (\boldsymbol{v}_f - \boldsymbol{v}_d)^2. \tag{27}$$

Here, $V(\boldsymbol{x})$ is a continuously differentiable function, and, for any $c > 0$, if the following inequality (28) holds, then $V(\boldsymbol{x})$ is a qualified CLF.

$$\inf_{\boldsymbol{u}\in\mathbb{R}} \left[\mathcal{L}_f V(\boldsymbol{x}) + \mathcal{L}_g V(\boldsymbol{x}) u + cV(\boldsymbol{x}) \right] \leq 0. \tag{28}$$

Next, we detail the hard constraints of this system. As previously mentioned, we aim for the distance between the UAVs and the UGV to remain within a specified range to ensure the high accuracy of data measured by the vision algorithm. We define these constraints as follows:

$$\begin{aligned} \boldsymbol{D} &\geq \tau_1 \boldsymbol{v}_f, \\ \boldsymbol{D} &\leq \tau_2 \boldsymbol{v}_f. \end{aligned} \tag{29}$$

Here, we propose two hard constraints: τ_1 is the time when the UAV catches up with the UGV and the the distance exceeds D_{max}, and τ_2 is the time when the UAV moves away from the UGV and distance narrows to within D_{min}. We consider the function $h_1(\boldsymbol{x}) = \boldsymbol{D} - \boldsymbol{v}_f \tau_1$ and $h_2(\boldsymbol{x}) = \boldsymbol{v}_f \tau_2 - \boldsymbol{D}$. Therefore, a candidate CBF can be expressed as

$$\begin{aligned} B(\boldsymbol{x}) &= -\log\left(\frac{h_1(\boldsymbol{x})}{1 + h_1(\boldsymbol{x})}\right), \\ C(\boldsymbol{x}) &= -\log\left(\frac{h_2(\boldsymbol{x})}{1 + h_2(\boldsymbol{x})}\right). \end{aligned} \tag{30}$$

We set $\gamma = 1$ in the following equation, thus verifying that $B(\boldsymbol{x})$ and $C(\boldsymbol{x})$ are qualified CBFs.

$$\begin{aligned} \inf_{u\in\mathbb{R}} \left[\mathcal{L}_f B(\boldsymbol{x}) + \mathcal{L}_g B(\boldsymbol{x}) u - \frac{\gamma}{B(\boldsymbol{x})} \right] &\leq 0, \\ \inf_{u\in\mathbb{R}} \left[\mathcal{L}_f C(\boldsymbol{x}) + \mathcal{L}_g C(\boldsymbol{x}) u - \frac{\gamma}{C(\boldsymbol{x})} \right] &\leq 0. \end{aligned} \tag{31}$$

From this, we build the CLF–CBF QP problem based on the constraints established above.

$$\begin{aligned} u^* &= \arg \min_{u=[u,\delta]^T} \frac{1}{2} u^T \boldsymbol{H}_{ag} u + \boldsymbol{F}_{ag}^T u \\ s.t. &\quad \begin{cases} \boldsymbol{A}_{clf} u \leq \boldsymbol{b}_{clf}, \\ \boldsymbol{A}_{cbf} u \leq \boldsymbol{b}_{cbf}, \end{cases} \end{aligned} \tag{32}$$

where

$$\begin{aligned} \boldsymbol{A}_{clf} &= \begin{bmatrix} \mathcal{L}_g V(\boldsymbol{x}) & -1 \end{bmatrix}, \\ \boldsymbol{A}_{cbf} &= \begin{bmatrix} \mathcal{L}_g B(\boldsymbol{x}) & 0 \\ \mathcal{L}_g C(\boldsymbol{x}) & 0 \end{bmatrix}, \end{aligned} \tag{33}$$

$$\begin{aligned} \boldsymbol{b}_{\text{clf}} &= \begin{bmatrix} -L_f V(\boldsymbol{x}) - cV(\boldsymbol{x}) \end{bmatrix}, \\ \boldsymbol{b}_{\text{cbf}} &= \begin{bmatrix} -\mathcal{L}_f B(\boldsymbol{x}) + \gamma \frac{1}{B(\boldsymbol{x})} \\ -\mathcal{L}_f C(\boldsymbol{x}) + \gamma \frac{1}{C(\boldsymbol{x})} \end{bmatrix}, \end{aligned} \tag{34}$$

$$\begin{cases} \mathcal{L}_g V(\boldsymbol{x}) = \frac{2(\boldsymbol{v}_f - \boldsymbol{v}_d)}{M}, \\ \mathcal{L}_f V(\boldsymbol{x}) = \frac{2F_r(\boldsymbol{v}_f - \boldsymbol{v}_d)}{M}, \end{cases} \tag{35}$$

$$\begin{cases} \mathcal{L}_g B(\boldsymbol{x}) = \frac{\tau_1}{M(1+h_1(\boldsymbol{x}))h_1(\boldsymbol{x})}, \\ \mathcal{L}_f B(\boldsymbol{x}) = \frac{-\tau_1 F_r / M + \boldsymbol{v}_l - \boldsymbol{v}_f}{(1+h_1(\boldsymbol{x}))h_1(\boldsymbol{x})}, \end{cases} \tag{36}$$

$$\begin{cases} \mathcal{L}_g C(\boldsymbol{x}) = \frac{-\tau_2}{M(1+h_2(\boldsymbol{x}))h_2(\boldsymbol{x})}, \\ \mathcal{L}_f C(\boldsymbol{x}) = \frac{\tau_2 F_r / M + \boldsymbol{v}_f - \boldsymbol{v}_l}{(1+h_2(\boldsymbol{x}))h_2(\boldsymbol{x})}. \end{cases} \tag{37}$$

In Equation (32), δ represents the relaxation factor. Setting δ to zero enforces precise exponential convergence at the rate c. Under these conditions, the constraints imposed by the CLF become more stringent, complicating the acquisition of suitable solutions for the QP equation. Meanwhile, the CLF converts the stability issue of the system into an optimization problem, aiming to minimize or satisfy specific performance metrics, articulated through a cost function. The construction of the CLF begins with partially linearizing the system using the feedback $u = F_r + M\mu$. Consequently , the cost function for this control mechanism is expressed as follows:

$$\mu\mu^T = |(F_r - u)/M|^2 = \frac{1}{M^2}\left(F_r^2 + u^2 - 2uF_r\right). \tag{38}$$

This can then be converted into Equation (39), where p_{sc} is the weight for the relaxation δ.

$$\boldsymbol{H}_{ag} = 2\begin{bmatrix} \frac{1}{M^2} & 0 \\ 0 & p_{sc} \end{bmatrix}, \quad \boldsymbol{F}_{ag} = -2\begin{bmatrix} F_r/M^2 \\ 0 \end{bmatrix}. \tag{39}$$

Therefore, we obtain the CBF–CLF Tracking system in Algoritm 2.

Algorithm 2: CBF–CLF Tracking Algorithm.

Input:
Initial state $\boldsymbol{x}(0)$;
Stop condition θ_m;
while $\theta < \theta_m$ **do**
- (a) UAV calculate relevant information (distance between UAV and UGV, poses of the Armor on UGV) through visual algorithm, and then run ICF algorithm;
- (b) Calculate the controller u by solving Equation (32);
- (c) Update UAV's horizontal velocity $\boldsymbol{v}_f$ by Equation (19);
- (d) Update UGV's velocity $\boldsymbol{v}_l$ using Equation (18);

end

3. Results

3.1. Vision-Based Localization

In the practical scenario where a UAV tracks a UGV, the UAV often needs to acquire the pose of the UGV for self-adjustment and decisionmaking. Computer vision algorithms are frequently employed to recognize the object's pose. However, due to the limitations in the sensor precision and the inherent design of these algorithms, the computed results often exhibit errors. This issue becomes particularly pronounced in distance measurements using the Perspective-n-Point (PnP) algorithm, where increased distance can cause objects to appear blurry or smaller in the image. Consequently, this leads to the visual features used for localization becoming less distinct or sparse, challenging the algorithm's ability to accurately match these feature points and affecting the precision of the pose estimation.

Given the aforementioned considerations, it is imperative to investigate the impact of distance variation on visual computation. With the knowledge of the camera's performance across different distances, it becomes possible to regulate the accuracy of the data acquisition by adjusting the observation distance, thus optimizing the system performance. The parameters used for experiment are shown in Table 1.

The camera was positioned at various distances and angles relative to the Armor, and the previously mentioned visual algorithm was employed to identify the light bars on the Armor and calculate their poses and angles. The distance between the camera and the Armor was computed, and the measurement error was then compared with the actual distance. The positions of the UAVs in the experiment are predetermined based on the distribution of the experimental variables, and each position is carefully measured and set. In this way, the error of the algorithm can be accurately calculated. The experimental setup is shown in Figure 5.

Table 1. Parameters for experiment.

Parameters Description	Value
Camera Model	MV-SUA133GC-T [1]
Calibration error	0.24 pixel
The height of Armor	125 mm
The width of Armor	150 mm

[1] MindVision USB 3.0 Industrial Camera, Shenzhen, China.

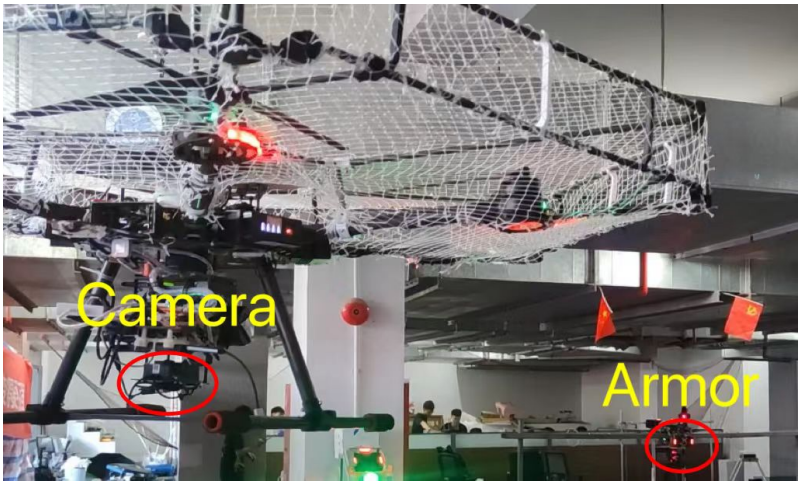

Figure 5. Experimental setup.

The experiment involved two variables: distance and angle. The distance varied from 3 m to 5 m, in increments of ten centimeters, resulting in 11 groups. The angle started with the camera facing the Armor at 0 degrees and ranged up to 40 degrees to the left, in increments of ten degrees, creating five groups. Considering the symmetry in the recognition of the left and right deviations, focusing solely on the left deviations was deemed sufficient for meeting the research objective. Consequently, this experiment comprised a total of 110 datasets.

Considering that an excessively large yaw angle could result in the loss of one side of the light bars, thereby making it impossible to form the visual features necessary for the Armor recognition, the experiment avoided overly large angles. Consequently, 40 degrees was selected as the maximum angle for this experiment.

As depicted in Figure 6, the error margin of the measurement data tends to widen as the distance increases. Within the range of 300 to 410 cm, the measurement error mostly fluctuates within 5 cm. Beginning at 420 cm, the measurement error starts to escalate, exhibiting significant deviations from previous values, with fluctuations between 10 and 15 cm. Consequently, we identify 410 cm as the threshold for accurate measurement. Beyond this distance, the camera pose estimation regarding the UAVs is deemed inaccurate, whereas, within this range, the estimation is considered to be reasonably accurate.

Meanwhile, the results in Figure 6 indicate that the deviations in angles have a relatively minor impact on the accuracy. The change in the yaw angle of the UAV relative to the UGV from 0 to 40 degrees has a negligible effect on the measurement data, which is theoretically justifiable. The algorithm functions by extracting the features from the upper and lower vertices of two light bars and then obtaining the rectangle corners through sorting. Consequently, the accuracy of the calculation is not dependent on the thickness of the light bars. Although an increase in the yaw angle may cause the captured light bars to appear thinner, it does not hinder the calculation process. However, excessively large yaw angles may lead to one side of the light bar becoming invisible, thereby creating an unsolvable scenario. Consequently, it is imperative to maintain the yaw angle below a critical threshold to ensure continuous observation and data integrity.

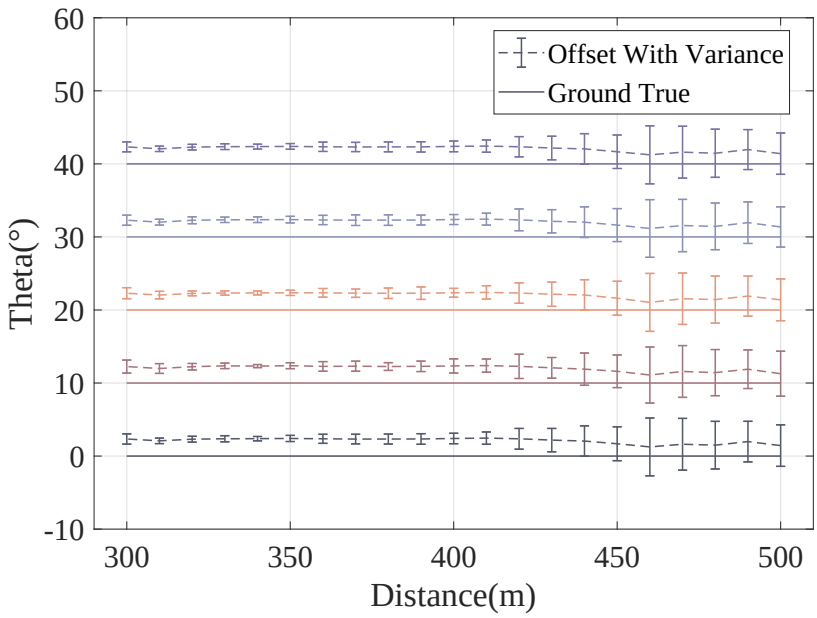

Figure 6. Distribution of measurement errors ranging at different distances and angles.

3.2. ICF

The tracking system described in this paper closely resembles a linear system. Regarding the algorithm performance, we posit that the ICF is somewhat less effective than the Centralized Kalman Filter (CKF). In the CKF, the sensor nodes collect a substantial amount of target information, facilitating the high-precision data processing of the accumulated information.

However, a significant characteristic of the CKF is its high-dimensional state space, which leads to a marked increase in computational complexity. This makes it challenging to fulfill the real-time requirements of navigation systems. Moreover, the fault tolerance of the CKF is relatively low. Should any node within the system fail, its impact spreads through the filter, affecting the other states and making the navigation information output of the combined system unreliable. Therefore, in practical applications, the ICF is preferred for mobile systems due to these considerations.

Based on the analysis above, if the average error curve of the ICF closely approximates that of the CKF, it is considered that the ICF has achieved a satisfactory performance. In this section, we evaluate the performance of the proposed ICF simulation in MATLAB 2023a and compare it with that of the CKF. The simulation of the tracking trajectory computed by the ICF algorithm is shown in Figure 7.

Initially, we investigate the influence of the number of cameras on the experimental results. Given the hardware limitations, the FOV of the cameras remains constant. To ensure comprehensive UGV observations, increasing the number of cameras and strategically placing them becomes imperative for accurate UGV tracking. It is essential that the UAV trajectory falls within the FOV of at least one camera; otherwise, it is considered unobservable and disregarded.

In this paper, we introduce two experimental variables: Process Covariance Q and Number of Consensus Iterations K. The UAVs are symmetrically distributed on both sides of the UGV. The input variables encompass the random refreshing of the UGV's position and an initial distance of 2 m between each UAV and the UGV. The velocity of the UGV is set to 1.5 m/s, with both the UAV's tracking speed and the UGV's leading speed undergoing slight random

variations. Consequently, the inputs for the simulation can be denoted as $\boldsymbol{x} = (x_i, y_i, v_x, v_y)$. The observation matrix $\boldsymbol{H}_i$ and state transition matrix $\boldsymbol{M}$ are defined as follows:

$$\boldsymbol{H}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{M} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{40}$$

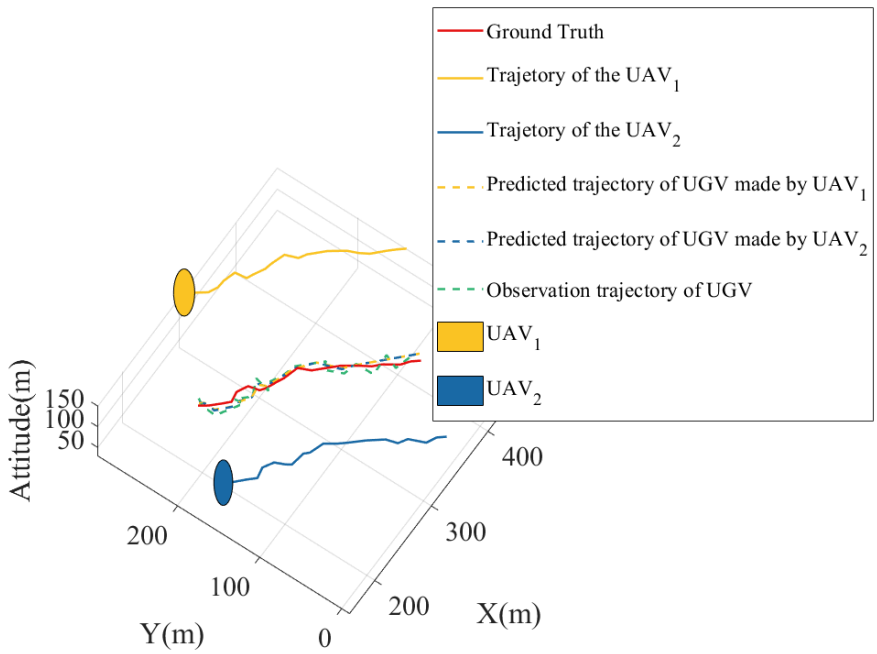

Figure 7. Simulation of the tracking trajectory computed by the ICF algorithm.

3.2.1. Process Covariance Q

The elements on the diagonal of the noise matrix represent the variance of measurement errors in the system's corresponding state variables. The magnitude of these diagonal elements significantly influences the filter's estimation accuracy of the system state. Larger diagonal elements usually indicate higher measurement noise in the corresponding state variables, rendering the system more sensitive to the impact of measurement noise. In this experiment, we have configured four sets of variables:

$$\begin{aligned} \boldsymbol{Q}_1 &= \mathrm{Diag}(0.01, 0.01, 1, 1), \\ \boldsymbol{Q}_2 &= \mathrm{Diag}(0.1, 0.1, 1, 1), \\ \boldsymbol{Q}_3 &= \mathrm{Diag}(1, 1, 1, 1), \\ \boldsymbol{Q}_4 &= \mathrm{Diag}(10, 10, 1, 1). \end{aligned} \tag{41}$$

In Figure 8, the CKF error curve shows that the error values across all the datasets are predominantly within ten, maintaining a consistently low level. For the ICF error curve, the first dataset displays the lowest error values, with the errors generally staying within 10 and peaking at 12 at most. Initially, the ICF error curve closely mirrors the CKF curve, indicating the efficient performance of the ICF. However, as the noise values on the diagonal escalate, there is a corresponding increase in the error values, leading to a divergence between the ICF and CKF curves. In the fourth dataset, the ICF's error values notably exceed 20, occasionally nearing 30, and the curve predominantly deviates from the CKF curve.

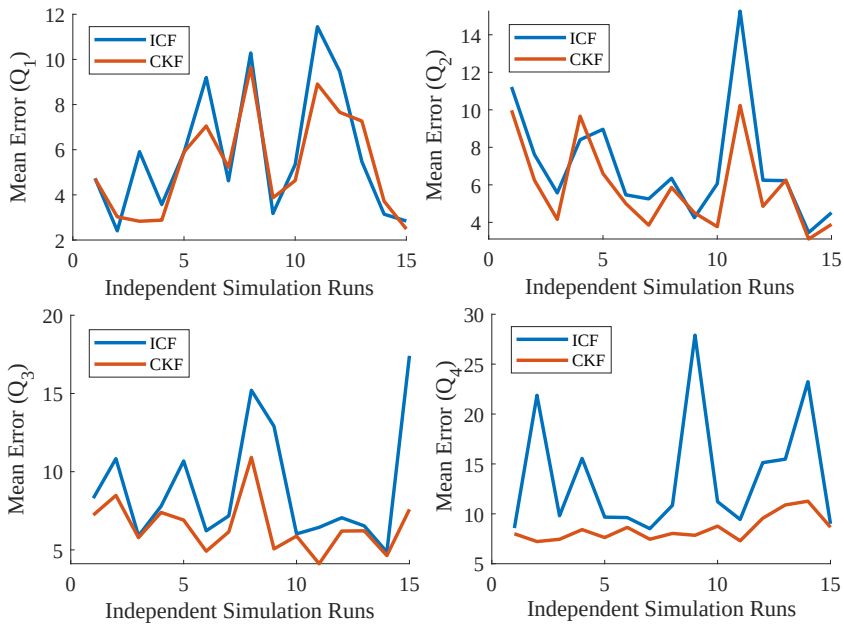

Figure 8. The mean error of independent simulation runs at different process covariance.

3.2.2. Consensus Iterations K

The parameter "total number of consensus iterations K" dictates the number of consensus iterations the algorithm undertakes throughout the process, significantly impacting the convergence and performance of the algorithm. The purpose of these consensus iterations is to ensure uniformity among the various nodes in a distributed system, allowing them to collaboratively estimate or deduce the system's state. As illustrated in Figure 9, the error curve of the ICF initially shows a significant deviation from the CKF error curve. However, as the number of iterations increases, specifically after $K = 4$, the error curve of the ICF effectively converges and becomes comparable to the CKF curve. Beyond this point, further increasing the number of iterations does not result in a marked improvement in the ICF's performance.

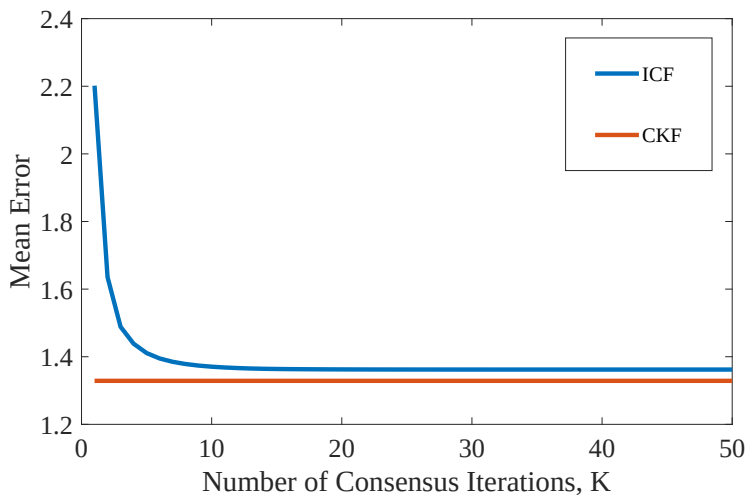

Figure 9. Mean error for different consensus iterations K.

A higher value of K denotes more iterations, which aids in bolstering the algorithm's stability in achieving consensus, but this can lead to prolonged computation times, consequently affecting the algorithm's convergence speed. Augmenting the number of consensus iterations could impose additional computational demands; hence, choosing an optimal K value

necessitates a balance between algorithm performance and computational efficiency. Given the dynamic nature of system states, the selection of K should also consider these dynamics. Larger K values are preferable in scenarios where the system states evolve gradually.

3.3. CBF–CLF

The findings from the preceding visual experiments suggest that maintaining a reasonable distance between UAVs and UGV is crucial to ensure the integrity and clarity of the Armor imaging. This experiment was simulated in MATLAB 2023a, and the parameters in this experiment are shown in Table 2. The initial velocities of the UAV and UGV are selected under the consideration of platform capabilities, safety, and efficiency.

Table 2. Parameters for experiment.

Parameters Description	Value	Parameters Description	Value
M	10 kg	v_f^0 **(case 1)**	1.0 m/s
f_0	0.1 N	v_l^0 **(case 1)**	1.5 m/s
$\boldsymbol{f_1}$	5 Ns/m	$\boldsymbol{v_f^0}$ **(case 2)**	0.9 m/s
$\boldsymbol{f_2}$	0.25 Ns^2/m	$\boldsymbol{v_l^0}$ **(case 2)**	0.5 m/s
$\boldsymbol{\tau_1}$	1 s	$\boldsymbol{p_{sc}}$	10
$\boldsymbol{\tau_2}$	3 s	$\boldsymbol{c}$	4
$\boldsymbol{\Delta t}$	1 s	$\boldsymbol{D_0}$	10
$\boldsymbol{v_d}$	v_l + 0.1 m/s	$\boldsymbol{g}$	9.81 m/s

In this paper, we consider the following two cases:

Case 1: the initial velocity of the UGV is greater than that of the UAV:

As depicted in Figure 10, the UGV tends to move away from the UAV. To close the distance gap while ensuring that the Armor imaging remains sufficiently large in the frame, the UAV increases its speed to catch up with the UGV. In the initial 0–10 s, the UAV's velocity, denoted as $\boldsymbol{v_f}$, exponentially increases as it swiftly narrows the gap with the UGV. Between 20 and 60 s, $\boldsymbol{v_f}$ gradually aligns with the UGV's velocity, denoted as $\boldsymbol{v_l}$, and achieves stability. After 60 s, $\boldsymbol{v_f}$ and $\boldsymbol{v_l}$ essentially equalize. Figure 11 (top) illustrates that the distance between the UAV and UGV stabilizes at a value not exceeding 2.4 m. Meanwhile, Figure 11 (bottom) presents the cumulative distribution function (CDF) of $\boldsymbol{D}/\boldsymbol{v_f}$, indicating that the UAV can catch up with the UGV within τ_1. This constraint naturally regulates the UAV's horizontal speed to ensure the continuous tracking of the UGV.

Case 2: the initial velocity of the UAV is less than that of the UGV:

As is shown in Figure 10, the UGV tends to move closer to the UAV. To increase the distance while preserving the integrity of the Armor imaging in the entire image and avoid scenarios where it becomes unrecognizable, the UAV decreases its velocity. In the initial 0–10 s, $\boldsymbol{v_f}$ exponentially decreases as the UAV slows down to match the UGV's pace. Between 20 and 60 s, the UAV's velocity gradually aligns with the UGV's velocity and stabilizes. After 60 s, $\boldsymbol{v_f}$ and $\boldsymbol{v_l}$ essentially become consistent, achieving the desired speed. Figure 11 indicates that the distance between the UAV and UGV stabilizes at a value not less than 1 m, while Figure 11 (bottom) displays the CDF of $\boldsymbol{D}/\boldsymbol{v_f}$, demonstrating that the UAV can catch up with the UGV within τ_2.

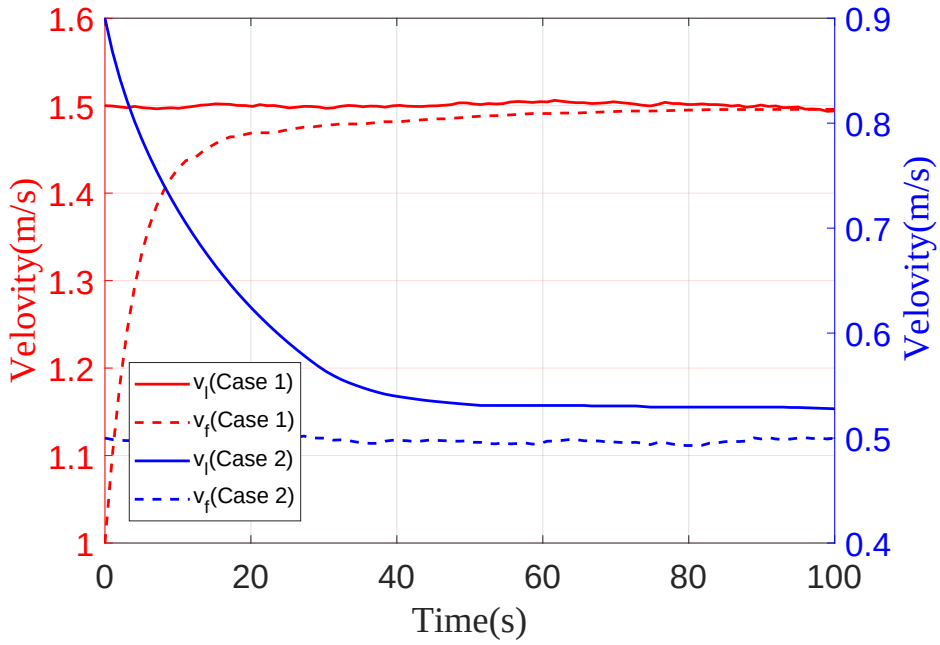

Figure 10. Velocity of UAV and UGV by conducting in two cases.

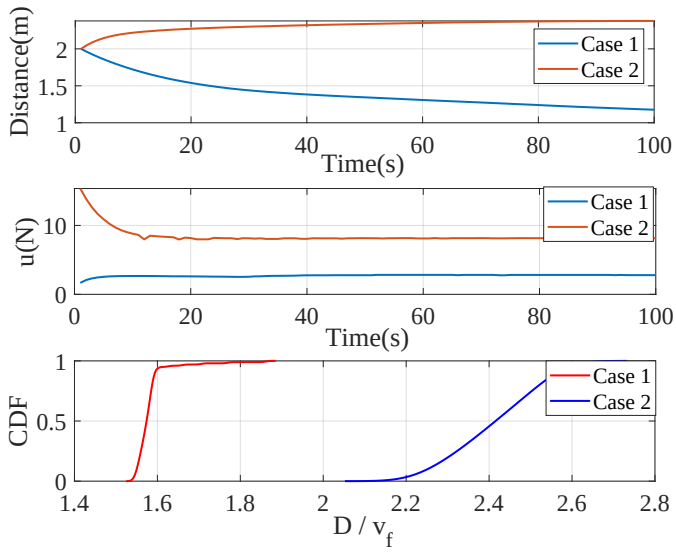

Figure 11. The distance between the UAV and UGV (**top**); control input (**middle**); the CDF of D/v_f for different τ (**bottom**).

4. Conclusions

This study explores the prospects of a decentralized control system through the integration of UAVs and UGVs in cooperative unmanned systems. The deployment of a decentralized control system, featuring the proposed CBF–CLF strategy, not only presents but also rigorously tests a method that substantially improves the operational safety and efficiency of the UAV swarm. Through the implementation of the ICF for the cooperative systems in real-time control adjustments, our proposed algorithm enables UAVs to operate independently. Additionally, the CBF–CLF strategy plays a crucial role in ensuring safety by automatically adjusting to keep the UAVs at safe distances from the UGVs. Extensive simulations across numerous real-world scenarios have proven the reliability of the proposed system and demonstrated its potential for extensive use in diverse UAV operational applications.

In future research, we could explore the application of anti-saturation fixed-time attitude tracking control in the cooperative systems to address the input saturation issues and ensure system stability. This approach integrates low-computation learning techniques to mitigate the impact of saturation and uncertainties. By incorporating anti-saturation fixed-time attitude tracking control into the cooperative systems, we can investigate its effectiveness in enhancing the control performance, robustness, and resilience against external disturbances.

Author Contributions: Conceptualization, B.O. and G.N.; methodology, B.O.; software, B.O. and G.N.; validation, F.L., B.O. and G.N.; formal analysis, B.O. and G.N.; investigation, F.L., B.O. and G.N.; resources, F.L.; data curation, F.L.; writing—original draft preparation, B.O., G.N. and F.L.; writing—review and editing, B.O., G.N. and F.L.; visualization, B.O. and F.L.; supervision, G.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fundamental Research Funds for the Central Universities of China under grant 20103237888.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicles
UGV	Unmanned Ground Vehicles
FOV	Field of View
PNP	Perspective-n-Point
DLT	Direct Linear Transform
CLF	Control Lyapunov Function
CBF	Control Barrier Function
ICF	Information-Weighted Consensus Filter
CKF	Centralized Kalman Filter
CDF	Cumulative Distribution Function

References

1. Alsamhi, S.H.; Shvetsov, A.V.; Kumar, S.; Shvetsova, S.V.; Alhartomi, M.A.; Hawbani, A.; Rajput, N.S.; Srivastava, S.; Saif, A.; Nyangaresi, V.O. UAV computing-assisted search and rescue mission framework for disaster and harsh environment mitigation. *Drones* **2022**, *6*, 154. [CrossRef]
2. Wang, Y.; Chen, W.; Luan, T.H.; Su, Z.; Xu, Q.; Li, R.; Chen, N. Task offloading for post-disaster rescue in unmanned aerial vehicles networks. *IEEE/ACM Trans. Netw.* **2022**, *30*, 1525–1539. [CrossRef]
3. De Almeida Barbosa Franco, J.; Domingues, A.M.; de Almeida Africano, N.; Deus, R.M.; Battistelle, R.A.G. Sustainability in the civil construction sector supported by industry 4.0 technologies: Challenges and opportunities. *Infrastructures* **2022**, *7*, 43. [CrossRef]
4. Niu, G.; Yang, Q.; Gao, Y.; Pun, M.O. Vision-based autonomous landing for unmanned aerial and ground vehicles cooperative systems. *IEEE Robot. Autom. Lett.* **2021**, *7*, 6234–6241. [CrossRef]
5. Motlagh, N.H.; Taleb, T.; Arouk, O. Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives. *IEEE Internet Things J.* **2016**, *3*, 899–922. [CrossRef]
6. Galar, D.; Kumar, U.; Seneviratne, D. *Robots, Drones, UAVs and UGVs for Operation and Maintenance*; CRC Press: Boca Raton, FL, USA, 2020.
7. Liang, X.; Zhao, S.; Chen, G.; Meng, G.; Wang, Y. Design and development of ground station for UAV/UGV heterogeneous collaborative system. *Ain Shams Eng. J.* **2021**, *12*, 3879–3889. [CrossRef]
8. Chen, J.; Zhang, X.; Xin, B.; Fang, H. Coordination between unmanned aerial and ground vehicles: A taxonomy and optimization perspective. *IEEE Trans. Cybern.* **2015**, *46*, 959–972. [CrossRef]
9. Gupta, A.; Fernando, X. Simultaneous localization and mapping (slam) and data fusion in unmanned aerial vehicles: Recent advances and challenges. *Drones* **2022**, *6*, 85. [CrossRef]
10. Miki, T.; Khrapchenkov, P.; Hori, K. UAV/UGV autonomous cooperation: UAV assists UGV to climb a cliff by attaching a tether. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), IEEE, Montreal, QC, Canada, 20–24 May 2019; pp. 8041–8047.

11. Bianchi, M.; Barfoot, T.D. UAV localization using autoencoded satellite images. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1761–1768. [CrossRef]
12. Wilson, A.; Kumar, A.; Jha, A.; Cenkeramaddi, L.R. Embedded sensors, communication technologies, computing platforms and machine learning for UAVs: A review. *IEEE Sens. J.* **2021**, *22*, 1807–1826. [CrossRef]
13. Wang, X.; Pan, H.; Guo, K.; Yang, X.; Luo, S. The evolution of LiDAR and its application in high precision measurement. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Beijing, China, 18–20 November 2019; Volume 502, p. 012008.
14. Sinfield, J.V.; Fagerman, D.; Colic, O. Evaluation of sensing technologies for on-the-go detection of macro-nutrients in cultivated soils. *Comput. Electron. Agric.* **2010**, *70*, 1–18. [CrossRef]
15. Minaeian, S.; Liu, J.; Son, Y.J. Vision-based target detection and localization via a team of cooperative UAV and UGVs. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *46*, 1005–1016. [CrossRef]
16. Yang, Q.; Shi, L.; Han, J.; Yu, J.; Huang, K. A near real-time deep learning approach for detecting rice phenology based on UAV images. *Agric. For. Meteorol.* **2020**, *287*, 107938. [CrossRef]
17. Wang, D.; Li, W.; Liu, X.; Li, N.; Zhang, C. UAV environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution. *Comput. Electron. Agric.* **2020**, *175*, 105523. [CrossRef]
18. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
19. Zhang, F.; Yu, J.; Lin, D.; Zhang, J. UnIC: Towards unmanned intelligent cluster and its integration into society. *Engineering* **2022**, *12*, 24–38. [CrossRef]
20. Behjat, A.; Manjunatha, H.; Kumar, P.K.; Jani, A.; Collins, L.; Ghassemi, P.; Distefano, J.; Doermann, D.; Dantu, K.; Esfahani, E.; et al. Learning robot swarm tactics over complex adversarial environments. In Proceedings of the 2021 IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS), Cambridge, UK, 4–5 November 2021; pp. 83–91.
21. Ruan, L.; Li, G.; Dai, W.; Tian, S.; Fan, G.; Wang, J.; Dai, X. Cooperative relative localization for UAV swarm in GNSS-denied environment: A coalition formation game approach. *IEEE Internet Things J.* **2021**, *9*, 11560–11577. [CrossRef]
22. Montecchiari, L. Hybrid Ground-Aerial Mesh Networks for IoT Monitoring Applications: Network Design and Software Platform Development. 2023. Available online: https://amsdottorato.unibo.it/10952/ (accessed on 20 June 2023).
23. Mammarella, M.; Comba, L.; Biglia, A.; Dabbene, F.; Gay, P. Cooperative agricultural operations of aerial and ground unmanned vehicles. In Proceedings of the 2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Trento, Italy, 4–6 November 2020; pp. 224–229.
24. Wu, Y.; Wu, S.; Hu, X. Cooperative path planning of UAVs & UGVs for a persistent surveillance task in urban environments. *IEEE Internet Things J.* **2020**, *8*, 4906–4919.
25. Phan, C.; Liu, H.H. A cooperative UAV/UGV platform for wildfire detection and fighting. In Proceedings of the 2008 IEEE Asia Simulation Conference-7th International Conference on System Simulation and Scientific Computing, Beijing, China, 10–12 October 2008; pp. 494–498.
26. Harik, E.H.C.; Guérin, F.; Guinand, F.; Brethé, J.F.; Pelvillain, H. UAV-UGV cooperation for objects transportation in an industrial area. In Proceedings of the 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 17–19 March 2015; pp. 547–552.
27. Arbanas, B.; Ivanovic, A.; Car, M.; Orsag, M.; Petrovic, T.; Bogdan, S. Decentralized planning and control for UAV–UGV cooperative teams. *Auton. Robot.* **2018**, *42*, 1601–1618. [CrossRef]
28. Kshemkalyani, A.D.; Singhal, M. *Distributed Computing: Principles, Algorithms, and Systems*; Cambridge University Press: Cambridge, UK, 2011.
29. Haeberlen, A.; Kouznetsov, P.; Druschel, P. PeerReview: Practical accountability for distributed systems. *ACM SIGOPS Oper. Syst. Rev.* **2007**, *41*, 175–188. [CrossRef]
30. Grocholsky, B.; Bayraktar, S.; Kumar, V.; Pappas, G. Uav and ugv collaboration for active ground feature search and localization. In Proceedings of the AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit, Chicago, IL, USA, 20–23 September 2004; p. 6565.
31. Jayavelu, S.; Kandath, H.; Sundaram, S. Dynamic area coverage for multi-UAV using distributed UGVs: A two-stage density estimation approach. In Proceedings of the 2018 Second IEEE International Conference on Robotic Computing (IRC), Laguna Hills, CA, USA, 31 January–2 February 2018; pp. 165–166.
32. Casbeer, D.W.; Beard, R. Distributed information filtering using consensus filters. In Proceedings of the 2009 IEEE American Control Conference, St. Louis, MO, USA, 10–12 June 2009; pp. 1882–1887.
33. Omotuyi, O.; Pokhrel, S.; Sharma, R. Distributed quadrotor uav tracking using a team of unmanned ground vehicles. In Proceedings of the AIAA Scitech 2021 Forum, Virtual, 11–15 & 19–21 January 2021; p. 0266.