



Article

# A Distributed Strategy for Target Tracking and Rendezvous Using UAVs Relying on Visual Information Only

Marco Andreetto <sup>1</sup>, Matteo Pacher <sup>1</sup>, David Macii <sup>2</sup>, Luigi Palopoli <sup>1</sup> and Daniele Fontanelli <sup>2,\*</sup>

<sup>1</sup> Department of Engineering and Computer Science, University of Trento, 38100 Trento, Italy; marco.andreetto@unitn.it (M.A.); matteo.pacher@polimi.it (M.P.); luigi.palopoli@unitn.it (L.P.)

<sup>2</sup> Department of Industrial Engineering, University of Trento, 38100 Trento, Italy; david.macii@unitn.it

\* Correspondence: daniele.fontanelli@unitn.it; Tel.: +39-0461-282080

† These authors contributed equally to this work.

Received: 23 August 2018; Accepted: 19 September 2018; Published: 21 September 2018



**Abstract:** This paper proposes a distributed target tracking solution using a team of Unmanned Aerial Vehicles (UAVs) equipped with low-cost visual sensors capable of measuring targets bearing information only. The team of UAVs moves along circular orbits and uses consensus-based distributed Kalman Filtering to identify the position of the target. We show that the centre of the orbit eventually converges to the target position using theoretical arguments and extensive simulation data. By using the same approach, we can solve the rendezvous problem: the team first scans an area in search of a target; as soon as one of the UAVs spots, the other components converge on the target position.

**Keywords:** distributed estimation; distributed tracking; distributed control; unmanned aerial vehicles

## 1. Introduction

The use of mini- and micro-UAVs (such as quad-rotors) is a promising solution to identify, monitor and (where required) track the position of a moving target [1]. The recent scientific literature has documented a wide range of applications, such as structural monitoring [2], fault detection occurring on airspeed sensors [3], and target tracking and encircling [4,5]. Our main goal in this paper is to show a strategy for the development of low cost and flexible UAV tracking applications using on board vision. This idea is certainly not novel [6–8], since visual sensors have considerable advantages: the limited cost, the ability to identify the target through image analysis and the possibility to exploit the same sensor for multiple purposes. Our main point is that in order to be of practical use, the localisation of the target has to be accurate and robust, e.g., with respect to visual occlusions. To meet these requirements, we use a team of UAVs that exchange information on the target, seek a consensus on its state and move in circular orbits around it.

It is a recognised fact that consensus distributed algorithm can be successfully applied to identify and track targets integrating the observation of multiple networked sensors [9,10]. The use of a cooperative distributed strategy to seek a stationary source has been championed by Li et al. [11], who developed a gradient ascent technique to drive the centre of the formation onto the target. Mariottini et al. [12,13] consider a scenario similar to the one presented in this paper, where a team of UAVs is required to track a mobile target and present a control technique relying on the use of 3D range finders. The same type of sensors are used by Olfati-Saber et al. [14], who propose a distributed Kalman filter, and by Rus et al. [15], who consider the UAVs as nodes of a mobile sensor network which have to agree on the presence of a certain event, e.g., a fire in the forest. We see some potential

drawbacks in the approaches that operate in 3D space using range information. First, specialised range sensors are quite expensive and difficult to integrate on micro- and mini- UAVs. Second, low cost visual sensors are known to produce noisy measurements when used to extract 3D range information. Third, control applications for UAVs 3D positioning relying on imprecise absolute information have important robustness limitations. Finally, the use of simplified dynamics for the UAVs as in [11–13] impedes a direct implementation of the solution on real hardware.

This paper proposes a distributed target tracking solution which solely exploits the bearing of the target with respect to the forward direction of each agent. This information is easily extracted from the image space using conventional cameras or thermal sensors (e.g., in wildlife monitoring applications) and it is arguably less affected by noise than the range estimation. Our approach is based on a two-phase distributed estimation algorithm combined with a distributed positioning control to increase noise immunity. More specifically, each UAV is controlled so that the target is driven into the centre of the images collected by the different UAVs. As a consequence, the target position can be estimated on the centre of the circular orbit of the UAVs. This algorithm produces correct results even with a single UAV but its performance improves with the number of elements of the team. This number can change in time, and the scalability is guaranteed by the adoption of a distributed coordination strategy with fast convergence. The capability to operate with a varying number of UAVs allows solving a rendezvous problem with the same algorithm.

The paper is organised as follows. In Section 2 we present some background material and give an overview of the solution. In Sections 3 and 4 we respectively discuss our solution for static and moving targets assuming that the UAVs move over the same circle. In Section 6 we discuss how to drive the position of far UAV into the circle and we show several simulations to prove the effectiveness of the approach in different conditions. Finally, Section 7 concludes the paper and outlines present and future work directions.

## 2. Background Material and Problem Formulation

We consider in this paper a team of  $n \geq 1$  UAVs (or agents) used to detect and track a target moving in a given environment by means of visual feedback. Each UAV is a quadrotor, for which we adopt the dynamic model introduced by Mahony et al. [16]. We borrow from the same paper the control scheme implementing a trajectory-following algorithm. As a result, we can concentrate on generating desired trajectories and safely assume that they will be followed. In Section 6, we will use explicitly the dynamic model to validate our results in a realistic simulation scenario.

The UAVs are equipped with a camera rigidly mounted on the chassis. The target detection process is performed by an off-the-shelf image processing algorithm. In addition, each UAV is equipped with a GPS receiver for absolute positioning, and with a magnetometer for yaw angle measurement.

UAVs communicate through a wireless link, which is used to set up a distributed control system. The robot team communication is associated with an undirected graph denoted as  $\mathcal{G}(t) = \{\mathcal{E}(t), \mathcal{V}(t)\}$ . At each time  $t$ ,  $\mathcal{V}(t) = \{1, 2, \dots, n\}$  is the vertex set, with each vertex representing a UAV, and  $\mathcal{E}(t) = \{\{i, j\} \mid i, j \in \mathcal{V}\}$  is the edge set, with each edge representing a communication link between two UAVs. The set of neighbours of agent  $i$  at time  $t$  is  $\mathcal{N}_i(t) = \{j \in \mathcal{V}(t) \mid \{i, j\} \in \mathcal{E}(t)\}$ . In practice,  $\mathcal{N}_i(t) = \{j \in \mathcal{V}(t) : \|\xi_i(t) - \xi_j(t)\| < C_r\}$ , where  $\xi_i(t) \in \mathbb{R}^3$  is the position of the  $i$ -th UAV,  $C_r$  is the communication range of the agents and  $\|\cdot\|$  the usual Euclidean norm.

### 2.1. Problem Formulation

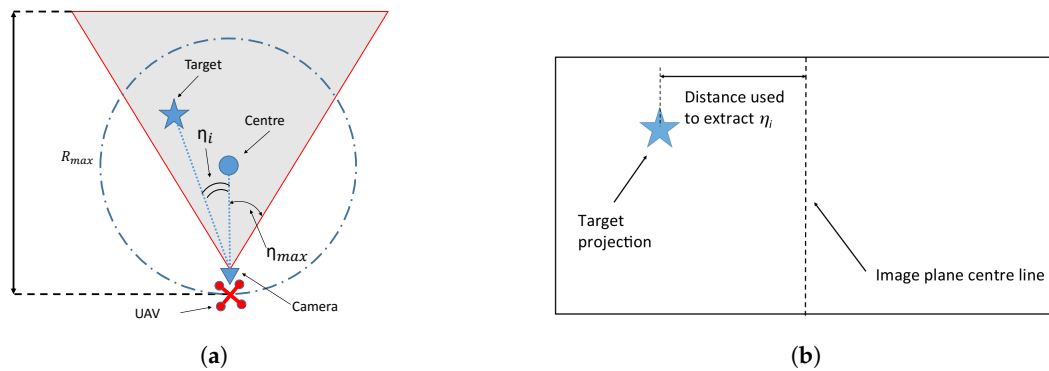
Let denote with  $p(t) = [p_x(t), p_y(t)]^T$  the actual position of the target on the ground with respect to  $\langle G \rangle$  and with  $\hat{p}(t)$  its estimate given by the team of robots. As depicted in Figure 1, the UAV camera points down and has a limited field-of-view (FOV). The extrinsic parameters of the camera are assumed to be known.

The quantity measured by each UAV is the bearing angle  $|\eta_i| \leq \eta_{max}$  (for  $i = 1, \dots, n$ ) under which the target is seen by the camera. The measurement of the bearing implies a line of sight ambiguity.

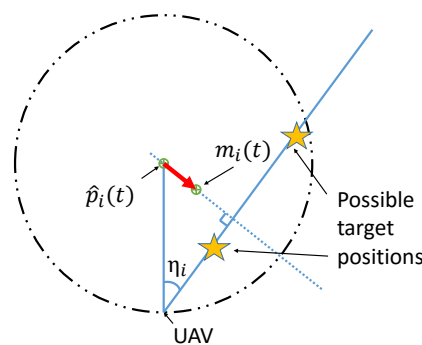
Indeed, given one single bearing measure  $\eta_i(t)$ , the target can be in any position on the line joining the  $i$ -th UAV position  $\xi_i(t) = [\xi_{x_i}(t), \xi_{y_i}(t), \xi_{z_i}(t)]^T$  with  $p(t)$  (see Figure 2). We define Equation (1) as the tracking error. Our goal is to minimise

$$e_p(t) = \|\hat{p}(t) - p(t)\|, \tag{1}$$

In order to have multiple views on the target taken from different positions, it is convenient to deploy the UAV team on closed orbits around the target [17]. The use of circular orbits has a considerable advantage: If the UAVs move along the same circle and if the target is kept in the centre of the orbit, then all the vision systems sense zero bearing angles. The converse is not always true, as discussed below. Moreover, the circle maximises the amount of information collected by a group of bearing sensor nodes, or by a single node taking multiple measures [17]. If we commit to circular orbits and define the centre of the circle of the  $i$ -th UAV as  $\hat{p}_i(t)$ , the problem can be formalised as follows: given a team of  $n \geq 1$  UAVs moving in an environment that are able to measure the bearing  $\eta_i$  of a given target by means of visual sensors, design a distributed algorithm able to drive all the UAVs on circular orbits having common centres  $\hat{p}_1(t) = \hat{p}_2(t) = \dots = \hat{p}_n(t) = \hat{p}(t)$  such that the tracking error  $e_p(t)$  in (1) is minimised.



**Figure 1.** (a) Bird’s eye view of the  $i$ -th drone placed on the circle. The shaded grey area represents the limited field-of-view (FOV) of the camera, while  $\eta_{max}$  is the maximum value of the measurable bearing. (b) Image seen from the  $i$ -th drone.



**Figure 2.** Representation of the local update of the estimate.

By solving this problem and assuming that the circular orbit has a known radius  $r$ , the position  $p(t)$  can be easily reconstructed, up to the tracking error (1), using the GPS position and heading (yaw angle) information of the UAVs. We assume that the image plane always points to the centre of the circle, e.g., using a gimbal support. For simplicity, but without loss of generality, we also

assume that the centre line of the image plane is aligned with the circle centre during the bearing angle measurement process, as depicted in Figure 1.

## 2.2. Solution Strategy

The proposed solution is organised in two steps. The UAVs are initially randomly deployed in the environment. In the first step, they start patrolling the area by moving independently from each other and at the same altitude. The patrolling strategy used in this phase is not actually relevant to this paper and the drones can be guided by any strategy ensuring collision avoidance (see [16] for an overview on quadrotor motion control). As soon as one of the UAV detects the target, it begins to take bearing measurements. At that time, the distributed strategy also begins and all the UAVs having the target within the communication range start sharing the information and eventually converge on the same orbit.

We will start our analysis by showing how the algorithm works for UAVs collecting bearing measures. In what follows, we will first present the solution for static targets (i.e., when  $\dot{p}(t) = 0$ ) and then the extension to dynamic targets (i.e., when  $\dot{p}(t) \neq 0$ ). Finally, we will show how the target localisation algorithm can be used to produce a *rendezvous* on the same circle.

## 3. Distributed Visual Servoing: Static Target

In this section we will show how the UAV team shifts the centre of their orbits towards a common value  $\hat{p}(t)$  that converges to the target position  $p(t)$  when  $\dot{p}(t) = 0$ . In this simplified case, since the target has a constant position, the tracking problem turns into a localisation problem. The results presented in this section subsumes the preliminary ideas presented in a previous paper [18] and addresses some critical limitations identified in the same paper. What is more, these results underpin the generalisation of the technique for moving targets presented in the next section.

In the considered scenario, the components of the team are initially deployed on the same circle (the algorithm tolerates small deviations from the circle). The algorithm operates iteratively, and each iteration consists of three phases, reported in Figure 3.

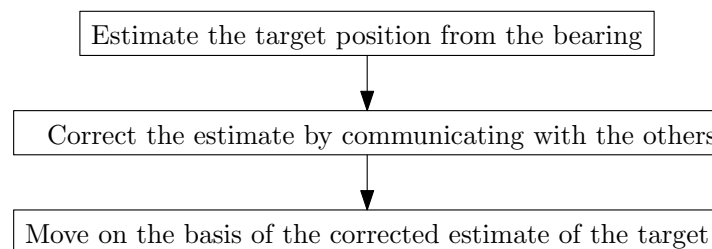


Figure 3. Workflow of the single agent.

### 3.1. Generation of Local Measurements

Local measurements are generated by each UAV having the target in sight by fusing the data measured through the camera with the previous estimate. If the  $i$ -th UAV detects the target within the FOV of its camera, the bearing offset  $\eta_i(t)$  at time  $t$  can be measured directly from the image. This is the case when the target is closer than a given distance  $R_{max}$  and the bearing error is smaller than  $\eta_{max}$  (see Figure 1).  $R_{max}$  and  $\eta_{max}$  depend on the features of the camera, the geometric configuration of camera and UAV, and on the height of the agents. Recalling that each agent aligns the principal axis of its camera with the radius of its orbit (see Figures 1 and 2),  $\eta_i(t)$  is given by the angle between  $r$  and the line-of-sight. By the right hand rule,  $\eta_i(t)$  is positive in counter-clockwise direction. We will denote with  $m_i(t)$  the *measure* of  $p(t)$  of the  $i$ -th UAV at time  $t$ . We compute the centre measure  $m_i(t)$  using the estimated centre of the orbit  $\hat{p}_i(t)$  and the bearing information  $\eta_i(t)$ , as

$$m_i(t) = \hat{p}_i(t) + l_{i,\perp} \text{sat}_{[-\lambda_i, \lambda_i]}(\kappa r \sin \eta_i(t)), \quad (2)$$

where  $\kappa > 0$  is a tuning gain,  $\text{sat}_{[-\lambda_i, \lambda_i]}(\alpha)$  is the saturation function

$$\text{sat}_{[-\lambda_i, \lambda_i]}(\alpha) = \begin{cases} \lambda_i, & \text{if } \alpha > \lambda_i, \\ \alpha, & \text{if } -\lambda_i \leq \alpha \leq \lambda_i, \\ -\lambda_i, & \text{if } \alpha < -\lambda_i, \end{cases} \quad (3)$$

$l_{i,\perp}$  is the unit vector orthogonal to the line-of-sight and directed towards it, and  $\lambda_i = \Lambda r \sin |\eta_i(t)| > 0$ ,  $0 < \Lambda < 2$  (see Figure 2). Notice that the measure  $m_i(t)$  is obtained by increasing the current centre position  $\hat{p}_i(t)$  by  $l_{i,\perp} \text{sat}_{[-\lambda_i, \lambda_i]}(\kappa r \sin \eta_i(t))$ . This increment coincides with the thick arrow in Figure 2. Its length is tuned by the gain  $\kappa$ , while its direction (orthogonal to the line-of-sight) is computed using the bearing measure  $\eta_i(t)$  and the GPS measure since the target position is unknown. Furthermore, if the UAV moves along  $l_{i,\perp}$  in order to have  $\hat{p}(t+1) = m_i(t)$ , the centre is made to approach the target provided that the increment (i.e., the length of the thick arrow in Figure 2) is small enough; this is the role of the saturation function.

### 3.2. Distributed Estimation

In the second phase, the UAVs propagate the measurements to their neighbours. Each UAV can then use this information to update the estimated position  $\hat{p}_i(t)$  of the target (which coincides with the desired position of the centre of its circular orbit). This step is carried out by using the distributed estimation algorithm proposed in [19]. This algorithm is applicable to any sensors network and produces at each node the maximum likelihood (ML) estimate, i.e., the optimal weighted least-squares (WLS) solution. In general, if  $y_i(t)$  denotes a (local) measure collected by the  $i$ -th agent, the algorithm executed by the  $i$ -th agent using the measures  $y_j(t)$ ,  $j \in \mathcal{N}_i(t)$ , returns  $y_{i,ML}$ , which is a local (i.e., known by the agent  $i$ ) approximation of the actual ML estimate  $y_{ML}$ . We adopt the Metropolis weighting method for this algorithm, which is proved to be very fast in terms of convergence rate (e.g., for the problem at hand, after five iterations the difference with respect to the true ML is negligible).

In general, the distributed ML estimate is always achieved at each node if the number of iterations approaches infinity, if the collection of communication graphs is jointly connected (consider that the set of  $r$  graphs  $\{\mathcal{G}_1, \dots, \mathcal{G}_r\}$  is jointly connected if the graph  $\mathcal{G} = \cup_{i=1}^r \mathcal{G}_i$  is connected), and if this connection occurs infinitely often. Roughly speaking, this condition implies only that in the long-term the graph is jointly connected. However, if some communication link fails permanently, the only convergence requirement is that the surviving agents belong to a connected graph. This very mild assumption is supposed to be valid for the problem at hand. An interesting property of the chosen algorithm is that even the intermediate estimates, i.e., before convergence, are unbiased. Another property is that the algorithm still works even if the number of UAVs that contributes to the estimate changes in time due to channel communication faults or FOV constraint violation.

In our application, the inputs of the estimation algorithm are the measures computed locally by means of (2). More precisely, the  $i$ -th agent uses the information  $m_j(t)$ ,  $j \in \mathcal{N}_i(t)$  and, after  $f \geq 5$  communication steps, the algorithm returns a common measure  $m_{i,ML}(t) \approx m_{ML}(t)$ ,  $\forall i = 1, \dots, n$ .

### 3.3. Position Update

After the application of the algorithm, the  $i$ -th agent updates the centre of its circular orbit using its best approximation  $m_{i,ML}(t)$  computed in a distributed way at Phase 2, i.e.,

$$\hat{p}_i(t+1) = m_{i,ML}(t). \quad (4)$$

The trajectory controller then moves the agent along an orbit with the new updated centre on the basis of the positioning strategy presented in Section 5.

### 3.4. Convergence Results

In order to solve the problem as defined in Section 2.2, the algorithm described above has to ensure that the centres  $\hat{p}_i(t)$  of the circular orbits of the UAVs will eventually converge to  $p(t) = p_0$ , i.e., static target. This is guaranteed by the following result.

**Theorem 1** (Target localisation). *Suppose that the target has a fixed position  $p(t) = p_0$  and that the cameras do not have limits on the FOV. If the centres of the orbits are updated as in (4),  $n \geq 3$  and the robots are in distinct positions, then*

$$\lim_{t \rightarrow \infty} \|\hat{p}_i(t) - p(t)\| = 0, \tag{5}$$

for all  $i \in \{1, \dots, n\}$ .

**Proof.** We first notice that if the target position  $p_0$  lies on the line passing through the current  $i$ -th centre  $\hat{p}_i(t)$  and the  $i$ -th agent, it follows that the local measurement law (2) is  $m_i(t) = \hat{p}_i(t)$  (see Figure 2).

Suppose now for simplicity that the WLS algorithm has converged, then, according to (4),  $\hat{p}(t) = \hat{p}_1(t) = \dots = \hat{p}_n(t) = m_{ML}(t - 1)$  is the common centre of the orbits. Then if all the local measurements return  $m_i(t) = \hat{p}_i(t) = \hat{p}(t)$  we have that  $\hat{p}(t) = p_0$ . If this is not the case, it follows from basic geometric arguments that each local measure  $m_i(t)$  is closer than  $\hat{p}(t)$  to the line-of-sight (Figure 2). Hence

$$\|m_i(t) - p(t)\| < \|\hat{p}(t) - p(t)\|, \tag{6}$$

i.e., each measure  $m_i$  is closer to the target than the current common centre  $\hat{p}(t)$ . Consider the convex envelop of all measures  $m_i(t)$ , which forms a polygon  $\Pi$  on the plane. Since  $m_{ML}(t) = \hat{p}(t + 1)$  is its centroid (because of the properties of the WLS), there always exists an agent  $j \in \{1, \dots, n\}$  such that

$$\|\hat{p}(t + 1) - p(t)\| \leq \|m_j(t) - p(t)\|. \tag{7}$$

Strictly speaking, (7) states that agent  $j$  proposes a measure that is worse than the ML estimate of the formation. Assume now that agent  $j$  proposes the worst measure; this is the farthest point of  $\Pi$  from the target and it is one of its vertexes. By combining (6) and (7), it finally results that

$$\|\hat{p}(t + 1) - p(t)\| \leq \|m_j(t) - p(t)\| < \|\hat{p}(t) - p(t)\|. \tag{8}$$

Equation (8) proves that the common centre estimate  $\hat{p}(t + 1)$  after the update at time  $t + 1$  is closer to the target position  $p(t)$  than  $\hat{p}(t)$ , i.e., the convergence of the estimates towards the actual value.

If the WLS has not converged yet, thanks to the property of intermediate estimates, Equation (7) can be replaced by  $\|\hat{p}_i(t + 1) - p(t)\| \leq \|m_j(t) - p(t)\|$ , i.e., the local result of the WLS algorithm  $\hat{p}_i(t + 1)$  still improves the measure  $m_j$  of the worst agent  $j$ . Then, Equation (8) is replaced by

$$\|\hat{p}_i(t + 1) - p(t)\| \leq \|m_j(t) - p(t)\| < \|\hat{p}_i(t) - p(t)\|, \tag{9}$$

hence, each centre asymptotically converges to the target position  $p(t) = p_0$ , as in the previous case.  $\square$

Notice that Theorem 1 is similar to the theorem reported in [18]. However, it is based on (2) that removes the constraint on the measures to be collected (i.e.,  $R_{min}$  in [18]).

For  $n \leq 2$ , Theorem 1 does not hold. Indeed, it may happen that  $p(t) \neq \hat{p}(t)$  lies on the line passing through the estimate  $\hat{p}(t)$  and the agent(s), but still  $\eta_i(t) = 0, i = 1, 2$ , which implies  $\hat{p}(t + 1) = \hat{p}(t)$ . Moreover, there is no guarantee that after the centre update law  $\hat{p}_i(t + 1) = m_{i,ML}(t), \forall i$ , there still exists at least one agent that senses the target due to the limited FOV: In that case the distributed algorithm would be at deadlock. To overcome both these limitations, a rotation around the circle centred in  $\hat{p}(t)$  is superimposed to the agents.



**Corollary 1.** Suppose that the target has a fixed position  $p(t) = p_0$ , and that  $n \geq 1$  UAVs move along the circle by taking at least 3 measures per revolution. If the centres of the orbits are updated via the distributed WLS algorithm and (4) is adopted, then

$$\lim_{t \rightarrow \infty} \|\hat{p}_i(t) - p(t)\| = 0, \quad (10)$$

for all  $i \in \{1, \dots, n\}$ , i.e., the target is tracked even if less than  $n = 3$  agents are used.

Corollary 1 ensures that even if there is only one UAV moving accordingly to the centre update law  $\hat{p}_1(t+1) = m_{1,ML}(t)$  that after the update does not sense the target at time  $t+1$ , we would have  $\hat{p}_1(t+2) = \hat{p}_1(t+1)$ . Since the centre does not change, its FOV (a triangle, according to Figure 1a) will sweep a circular area per revolution centred in the orbit of the UAV and larger or equal to the surface covered by the circular orbit. Since after the update the centre will be closer to the target, there will be a finite time in which the target will be inside the FOV in one revolution. Of course, to avoid singular configurations ( $\hat{p}_1(t) \neq p(t)$  and  $\eta_1(t) = 0$  from different locations on the circle), it is sufficient to have at least 3 different measures per revolution. Holding for one UAV, it follows that  $n > 1$ .

Notice that the results of Theorem 1 and Corollary 1 still hold even if the measurements based on (2) are corrupted by some additive white and normally distributed noise  $\varepsilon_i(t)$ .

#### 4. Distributed Visual Servoing: Dynamic Target

For the case of moving targets, we assume to use the same measurement model (2) used for static targets. The main difference is that  $\dot{p}(t) \neq 0$ , which implies a possible inefficiency of a non-Bayesian approach such as the distributed WLS. Our solution is based on the iterative distributed implementation of a distributed Kalman filter (DKF) for a generic sensors network in [10], shortly described as follows. Suppose we need to estimate the state  $x$  of a dynamic system

$$x(t+1) = A(t)x(t) + B(t)u(t) + G(t)w(t), \quad (11)$$

where  $x \in \mathbb{R}^m$ ,  $A \in \mathbb{R}^{m \times m}$  is the dynamic matrix,  $u \in \mathbb{R}^p$  is the input,  $B \in \mathbb{R}^{m \times p}$  is the input matrix,  $w \in \mathbb{R}^p$  is the process noise and  $G \in \mathbb{R}^{m \times p}$ .  $A$ ,  $B$  and  $G(t)$  may be time varying. The sensor  $i$  collects a noisy measure  $z_i$ :

$$z_i(t) = H_i(t)x(t) + v_i(t), \quad (12)$$

where  $z \in \mathbb{R}^q$ ,  $H_i$  is the gain of the  $i$ -th sensor and  $v_i \in \mathbb{R}^l$  is the  $i$ -th measurement noise.  $w$  and  $v_i$  are supposed zero-mean Gaussian noises. The DKF executed by the  $i$ -th agent takes as input the measure  $z_i(t)$  and produces a filtered estimate of the state  $\hat{x}_i(t)$ , by exchanging the local estimates through the communication channel. Again, if  $t \rightarrow +\infty$ , the common minimum variance estimate  $\hat{x}_1(t) = \hat{x}_2(t) = \dots = \hat{x}_n(t) = \hat{x}(t)$  is produced by the DKF.

##### Target Tracking

Theorem 1 holds only for a static target since relation (6) does not hold if  $\dot{p}(t) \neq 0$ . In this case, the target tracking can be obtained by combining the WLS distributed filter with the DKF just introduced to estimate both the target position and velocity [20]. We model the target as a generic second-order system with state  $x(t) = [p(t)^T, \dot{p}(t)^T]^T$  in (11). However, in the simulations, the real target dynamics will be more complex than the second-order system implemented in the filter (see Section 6.2). For observability reasons, the DKF needs at least the measurement of the target position  $p(t)$  that should be used in (12). This information cannot be extracted from a single measurement, but it can be reconstructed using the distributed paradigm. Thereby, our solution is to use the results of the target localisation described in Section 3 as measurement. The rationale of this choice is that (7) is still verified in the presence of a moving target, since the output  $m_{i,ML}(t)$  of the distributed WLS algorithm improves the measure of the worst agent and reduces the uncertainty contribution  $\varepsilon_i(t)$  associated

with measurement model (2). The tracking algorithm can then be thought as the cascade of two stages, i.e., *approaching* and *tracking*, as explained below.

In the **approaching** stage, first static target localisation is performed by applying the distributed WLS on a predefined number of sensor readings.

In the following **tracking** stage at each time step  $t$ , the agent  $i$  computes the following operations:

1. Compute the measurement  $m_i(t)$  from  $\eta_i$  using (2);
2. Execute one step of the WLS algorithm with  $m_i(t)$  as an input and produce  $m_{i,ML}(t)$ ;
3. Run the DKF algorithm using as inputs  $z_i(t) = m_{i,ML}(t)$  returned by the WLS. The output is the estimate  $\hat{x}_i(t)$  of the target state (position and velocity);
4. Run again one step of the WLS algorithm using the states  $\hat{x}_i(t)$  estimated by the DKF in the previous step as inputs ( $y_i$  computed in Phase 2 of the algorithm in Section 3). In accordance with the notation introduced in Section 3, the output  $\hat{x}_{i,ML}(t)$  is a local approximation of the ML estimate between the inputs  $\hat{x}_j(t)$ ,  $j = 1, \dots, n$ ;
5. The updated centre position and velocity are then given by  $[\hat{p}_i(t+1)^T, \hat{p}_i(t+1)^T]^T = \hat{x}_{i,ML}(t)$ .

**Remark 1** (Interpretation of the approaching stage). *The approaching stage consists of a first rough localisation of the target. The roughness of the localisation process comes from the fact the target is moving. Indeed, by direct Euler integration of the second order system in (11) with sampling time  $\Delta_t$ , we have*

$$p(t+1) = p(t) + \Delta_t \dot{p}(t). \quad (13)$$

Moreover, we define  $\rho(t) \triangleq \|\Delta_t \dot{p}(t)\|$ . From (13), it follows that  $\|\hat{p}_i(t+1) - p(t+1)\| \leq \|\hat{p}_i(t+1) - p(t)\| + \rho(t)$ , and, since from Theorem 1 we have that relation (9) still holds true, we can safely state that if condition  $\|\hat{p}_i(t+1) - p(t)\| + \rho(t) \leq \|\hat{p}_i(t) - p(t)\|$ , is verified, we have that  $\|\hat{p}_i(t+1) - p(t+1)\| \leq \|\hat{p}_i(t) - p(t)\|$ . In other words, if the motion of the target is negligible with respect to the displacement of the orbit centre generated by the algorithm, the common centre will converge on the target. In such a case (which is experimentally verified in Section 6), if the target is far from the UAVs common orbit centre, the target is approached correctly.

**Remark 2** (Interpretation of the tracking stage). *In the tracking stage, a DKF is used to handle the motion of the target. The input  $z_i = m_{i,ML}(t)$  is computed by running one step of the localisation algorithm in Section 3.4, which is considered then as a single distributed sensor. Then one step of the DKF is carried out. It has to be noted that the estimates  $\hat{x}_i(t)$ ,  $\forall i$ , can be fairly different. To ensure that the centres of the orbits are close, i.e.,  $\hat{p}_1(t) \approx \hat{p}_2(t) \approx \dots \approx \hat{p}_n(t)$ , the DKF is finally followed by another step of WLS.*

A final remark is on whiteness of the noise that affects the measures of the DKF, i.e., the independence property of the Gaussian process generating the measurement noise  $v_i(t)$  in (12). We notice that the autocorrelation of the noise tends to a Dirac delta (white noise) if the gain  $\kappa$  is sufficiently large, i.e., greater than 5. The noise has been obtained by combining the noise  $\varepsilon_i(t)$  that affects the values based on (2), i.e., yaw and position of the robot based on magnetometer and GPS receiver, respectively, and the bearing error  $\eta_i$  sensed by the camera.

## 5. Distributed Positioning

In the previous sections, we have discussed a distributed strategy to shift the centre of a formation of robots deployed on a circle so it reaches a target. As regards the motion of the UAV on the circular orbit, Zhao et al. [17] have shown that the optimal condition requires that the UAV sensors are equally spaced on the circle. To obtain this result, we can combine the translation motion of the centre of the circle with a rotational motion of the agents [18]. The  $i$ -th UAV computes its position set-point by using the position and velocity of pairs of neighbour UAVs to compute the correct rotation on the circle, and then by superimposing this circular motion to a translation towards the estimated centre  $\hat{p}_i(t)$ .



For a static target or, equivalently, for the approaching phase, only the position of the centre given by the local update rule in (4) is used. For dynamic targets during the tracking phase, the translation is computed using both the estimated position  $\hat{p}_i(t)$  and velocity  $\dot{\hat{p}}_i(t)$  of the target, which are available through the application of the DKF. After a motion vector is computed, the execution of the motion is delegated to the trajectory-following algorithm shipped with the UAV [16].

### Rendezvous

The problem of how to attract on a circular orbit the UAVs that are initially far away can be thought of a problem, in which the agents have to eventually converge to the same circular orbit.

After an initial phase in which the UAVs patrol the area independently, suppose that the  $i$ -th UAV spots the target. The application of the algorithm described earlier guarantees that UAV  $i$  will evolve on a circular orbit with the target in its centre. Consider a generic element  $j$  of the team, with  $j \neq i$ . Depending on the communication range, on the distances and on the presence of occlusions, we can have at each step three situations (the case in which UAV  $j$  does not see the target and does not communicate with  $i$  is not relevant since  $j$  has just to continue the search):

1. UAV  $j$  does not see the target but it communicates with UAV  $i$ ;
2. UAV  $j$  sees the target, but does not communicate with UAV  $i$ ;
3. UAV  $j$  sees the target and communicates with UAV  $i$ .

In the first case, UAV  $i$  communicates the circle and UAV  $j$  executes a point-to-point motion using its navigation algorithm. In the second case, UAV  $j$  evolves toward the circular orbit independently (just as UAV  $i$  does). The convergence toward the rendezvous is in this case “physical” and is pivoted by the target. In the third case, the UAV takes advantage of the consensus strategy ensuring a faster and more robust converge toward the goal. The rendezvous between UAVs  $i$  and  $j$  is guaranteed as long as at each step one of these three conditions is verified. Moreover, as long as the communication range is greater than or equal to the orbit radius, the system will asymptotically converge toward a situation in which the third scenario is the most frequent.

## 6. Simulation Results

The performance of the proposed algorithm for both static and moving target localisation is presented in this section. In all the simulations reported, a variable number  $n$  of drones has been adopted. Realistic parameters have been used for the UAV dynamic and sensing systems. The maximum velocity of each agent is  $\dot{\zeta}_i^{max} = 13$  m/s, which is the declared maximum velocity for a commercial *Parrot* UAV [21]. Moreover, the dynamic model proposed by Mahony et al. [16] has been used. The FOV maximum aperture is set to  $\eta_{max} = \pm 30^\circ$ . The UAV positions are measured by standard GPS receivers, with a zero-mean, normally distributed uncertainty with standard deviation equal to 2 cm, as reported in [22].

### 6.1. Tracking Performance

#### 6.1.1. Set-Up

In the first set of simulations, we suppose that, at the beginning, the UAVs are randomly deployed on a circle, whose centre is located 18 m away from the target. The bearing noise affecting the camera measures is supposed to exhibit a Gaussian distribution with zero mean and standard deviation equal to  $1^\circ$ . This also comprises the magnetometer uncertainty, as described in [23]. The wireless communication range is set to 60 m. The simulations are run with a time step of 0.05 s. At each time step, each UAV sends 3 messages and receive all the messages sent by its neighbours. This implies that the distributed estimation algorithm described in Section 4 is run once per time step. The aim of these simulations is to compare the tracking performance of the WLS algorithm in Section 3.4 (hereafter called static) with the algorithm described in Section 4 (hereafter called dynamic). To this

purpose, a high sampling rate was used. The performance index is the average steady-state estimation uncertainty, i.e., the steady state Root Mean Squared Error (RMSE). This is obtained by averaging the results of 50 different simulations. Both the static and the dynamic localisation algorithms have been tested with a moving target whose dynamic is given by

$$\dot{x}(t) = \begin{bmatrix} 0 & I_2 \\ 0 & 0 \end{bmatrix} x(t) - 4d\pi^2\omega^2 \begin{bmatrix} 0 \\ I_2 \end{bmatrix} u(t), \tag{14}$$

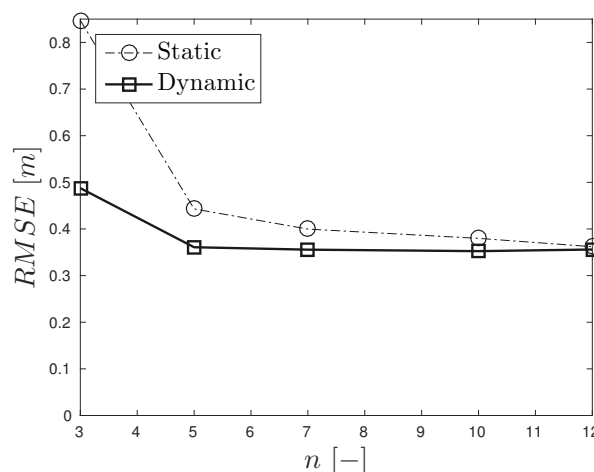
where  $d = 20$  m,  $\omega = 0.0285$  Hz,  $I_2$  is the bi-dimensional identity matrix, and

$$u(t) = \begin{bmatrix} 4 \sin(4\pi\omega t) \\ \sin(2\pi\omega t) \end{bmatrix}. \tag{15}$$

The initial condition is  $x(0) = [0, 0, 0, 0]^T$ . This way, a challenging periodic “8” shaped trajectory is obtained. The values of parameters  $d$  and  $\omega$  are based on the maximum velocity of the agents. The Kalman filter has been implemented by supposing constant target acceleration in the model, and initialised by setting a standard deviation of the acceleration noise equal to  $0.01$  m/s<sup>2</sup>.

### 6.1.2. Analysis

Figure 4 shows the RMSE between the common centre position  $\hat{p}(t)$  and the real position of the target as a function of the number of drones, i.e., the RMSE of  $\|\hat{p}(t) - p(t)\|$  for a time greater than 20 s.



**Figure 4.** Root mean square error (RMSE) of the tracking error for the static (dash-dotted) and dynamic (solid) algorithms as a function of the number of drones.

As expected, the larger the number of UAVs, the better the performance. The main outcomes are twofold. First, it is shown that the static algorithm works reasonably well for manoeuvring targets, hence its use in the *Approaching stage* of dynamic algorithm Section 4 is reasonable. Second, the effect of the DKF is remarkable since the same precision achieved using 12 drones for the static algorithm is attained with only 5 drones when the DKF is employed. As far as convergence is concerned, we have that the RMSE reaches the steady state in less than 4 s for  $n = 3$ , and in less than 1 s for  $n = 12$  with both algorithms.

To better highlight the advantages of the proposed algorithm, we have additionally set  $u(t) = 0$  with initial condition  $x(0) = [0, 0, v, v]^T$ , where  $v \in [0, 8]$  m/s. This way, a constant velocity, linear trajectory model has been simulated. Again, for each possible values of  $v$ , the test was repeated 50 times using  $n = 5$  drones. As expected, the larger the velocity of the target, the better the performance of the dynamic algorithm with respect to the static one, which is a direct consequence of the adoption of the

dynamic model in (11). For numerical comparison, the static algorithm yields the RMSE ranging from 0.27 m for  $v = 0$  to 0.9 m for  $v = 8$  m/s, while, adopting the dynamic one, the RMSE grows almost linearly from 0.25 m for  $v = 0$  to 0.62 m for  $v = 8$  m/s.

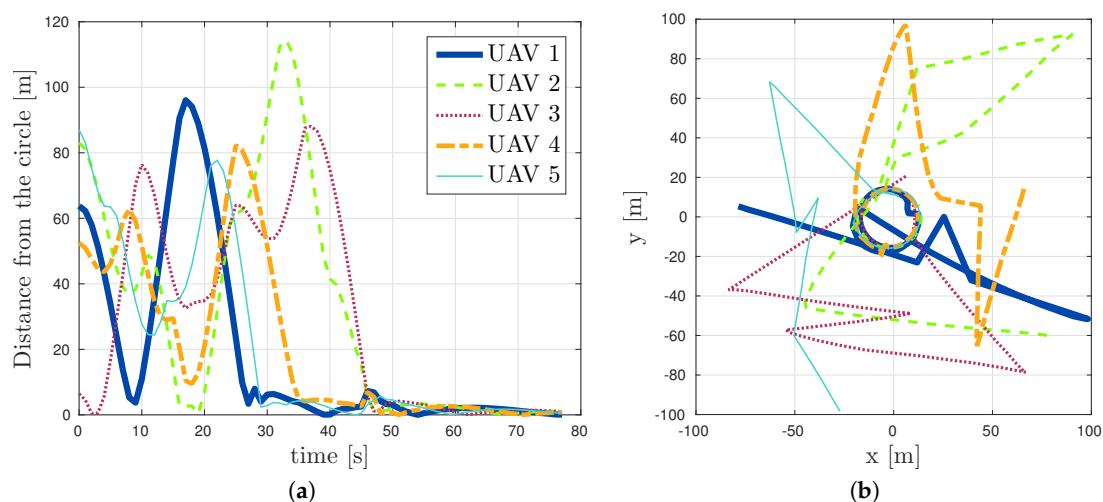
## 6.2. Rendezvous

### 6.2.1. Set-Up

The proposed distributed strategy has been applied to solve the rendezvous problem as well. The bearing noise and the magnetometer noise are supposed to exhibit a Gaussian distribution with zero mean and standard deviation equal to  $1^\circ$  and  $0.5^\circ$ , respectively. With such a choice for the noises, we have verified through the  $\chi^2$  test on  $10^5$  samples that the noises  $\varepsilon_i(t)$  affecting (2) and the measurement noise  $v_i(t)$  in (12) are Gaussian, in accordance with the DKF assumptions. The FOV maximum distance is set to  $R_{max} = 30$  m, while the radius adopted for the circular orbit is  $r = 15$  m. The wireless communication range is set to 30 m, which is a reasonable value for short range radio modules (e.g., based on IEEE 802.15.4). In this case, the simulations time step is 1.0 s. At each time step, each UAV is supposed to send up to 3 messages and to receive all the messages sent by its neighbours. This implies that the distributed estimation described in Section 4 may run once per time step. This sampling rate is quite low and can be easily achieved in practice. The UAVs are initially randomly deployed in the patrolled area (which is  $200 \times 200$  m<sup>2</sup>) and they move to random positions (via point-to-point motion) inside this area looking for the target.

### 6.2.2. Analysis

Figure 5a reports the time evolution of the distances between each UAV and the circle centred in the target, which is supposed to have the dynamic given by (11) with  $x(0) = [0, 0, 0, 0]^T$ ,  $u(t) = 0$ . This means that static target is placed in the origin of the chosen planar reference frame. The Kalman filter has been initialised like in the previous simulations.



**Figure 5.** (a) Distance of the UAVs from the circle centred on the fixed target in position  $p(t) = [0, 0]^T$ . (b) Converging trajectories of the agents in the rendezvous with fixed target (the legend in (a) holds).

It is evident that convergence time is longer than in the previous case due to the searching phase. However, all the UAVs converge eventually to the same location, hence solving the rendezvous problem. Moreover, during the transient phase it may happen that some distances decrease and increase continuously. This behaviour is a direct consequence of the UAVs limited FOV. In fact, it may happen that a robot passes through the desired circle without detecting the target. Figure 5b pictorially

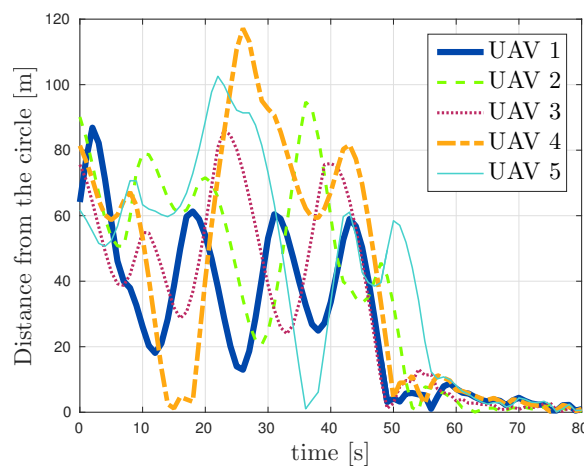
depicts the UAVs converging trajectories. We finally present the rendezvous problem for a target moving in man-like form. The model chosen for target motion is given by

$$\dot{x}(t) = \begin{bmatrix} 0 & I_4 \\ 0 & 0 \end{bmatrix} x(t) - \begin{bmatrix} 0 \\ I_2 \end{bmatrix} u(t), \quad (16)$$

where  $I_4$  is the  $4 \times 4$  identity matrix and

$$u(t) = \begin{bmatrix} -\tilde{\alpha} x_5 + w_5(t) \\ -\tilde{\alpha} x_6 + w_6(t) \end{bmatrix}, \quad (17)$$

with  $\tilde{\alpha} > 0$ . Noise terms  $w_5(t)$  and  $w_6(t)$  exhibit zero-mean normal distributions, while  $x_k$  denotes the  $k$ -th component of state vector  $x$ . In the simulations we set  $\tilde{\alpha} = 10^{-2}$ , while the standard deviation of  $w_5(t)$  and  $w_6(t)$  are set equal to  $0.1 \text{ m/s}^3$ . The initial condition is  $x(0) = [0, 0, 1, 1, 1, 1]^T$ . This dynamic is a well established model for manoeuvring targets [24–26], which is also used to model human motion [27,28]. The model implemented in the Kalman filter is the previous double integrator. Again, all distances converge to zero, as expected even in this more challenging scenario (see Figure 6).



**Figure 6.** Distance of the UAVs from the circle centred on the target moving in manlike form (16).

## 7. Conclusions

A solution based on a distributed and fully decentralised control of a team of UAVs to optimise their ability to detect and to locate an unknown moving target is presented in this paper. The proposed algorithm relies on both a very robust and simple bearing-only visual servoing technique and a consensus scheme fusing a Maximum Likelihood algorithm and a Kalman filter. The simulation results clearly show that the proposed solution is viable and effective in any possible initial configuration of the drones with respect to the target and it works efficiently also for the rendezvous problem. The algorithm accuracy depends on the number of agents. In particular, for a given application-dependent steady-state positioning uncertainty, the minimum number of agents to be used can be determined. In future, the proposed solution could be extended to the case of fixed-wing drones and it could be implemented on real robotics agents.

**Author Contributions:** All the Authors have equally contributed to the paper conceptualization, methodology; formal analysis; writing—original draft preparation; writing—review and editing.

**Funding:** The research activity described in this paper has received funding from the University of Trento within the project “D-FAB - Drones for Finding Avalanche-Buried”.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Fravolini, M.; Pastorelli, M.; Pagnottelli, S.; Valigi, P.; Gururajan, S.; Chao, H.; Napolitano, M. Model-based approaches for the airspeed estimation and fault monitoring of an Unmanned Aerial Vehicle. In Proceedings of the 2012 IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS), Perugia, Italy, 28 September 2012.
2. Yoon, H.; Shin, J.; Spencer, B.F., Jr. Structural Displacement Measurement using an Unmanned Aerial System. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 183–192. [[CrossRef](#)]
3. Gomez-Balderas, J.; Flores, G.; García Carrillo, L.; Lozano, R. Tracking a Ground Moving Target with a Quadrotor Using Switching Control. *J. Intell. Robot. Syst.* **2013**, *70*, 65–78. [[CrossRef](#)]
4. Owen, M.; Yu, H.; McLain, T.; Beard, R. Moving ground target tracking in urban terrain using air/ground vehicles. In Proceedings of 2010 IEEE Globecom Workshops, Miami, FL, USA, 6–10 December 2010. [[CrossRef](#)]
5. Iskandarani, M.; Hafez, A.; Givigi, S.; Beaulieu, A.; Rabbath, C. Using multiple Quadrotor aircraft and Linear Model Predictive Control for the encirclement of a target. In Proceedings of the 2013 IEEE International Systems Conference (SysCon), Orlando, FL, USA, 15–18 April 2013. [[CrossRef](#)]
6. Doitsidis, L.; Weiss, S.; Renzaglia, A.; Achtelik, M.W.; Kosmatopoulos, E.; Siegwart, R.; Scaramuzza, D. Optimal Surveillance Coverage for Teams of Micro Aerial Vehicles in GPS-denied Environments Using Onboard Vision. *Auton. Robot.* **2012**, *33*, 173–188. [[CrossRef](#)]
7. Qadir, A.; Neubert, J.; Semke, W.; Schultz, R. On-board visual tracking with unmanned aircraft system (uas). In Proceedings of the Infotech@Aerospace 2011, St. Louis, MO, USA, 29–31 March 2011.
8. Fasano, G.; Accardo, D.; Tirri, A.E.; Moccia, A.; De Lellis, E. Morphological filtering and target tracking for vision-based UAS sense and avoid. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014.
9. Carli, R.; Chiuso, A.; Schenato, L.; Zampieri, S. Distributed Kalman filtering based on consensus strategies. *IEEE J. Sel. Areas Commun.* **2008**, *26*, 622–633. [[CrossRef](#)]
10. Olfati-Saber, R. Kalman-Consensus Filter: Optimality, stability, and performance. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009. [[CrossRef](#)]
11. Li, S.; Kong, R.; Guo, Y. Cooperative distributed source seeking by multiple robots: Algorithms and experiments. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 1810–1820. [[CrossRef](#)]
12. Morbidi, F.; Mariottini, G.L. Active target tracking and cooperative localization for teams of aerial vehicles. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 1694–1707. [[CrossRef](#)]
13. Gurcuoglu, U.; Puerto-Souza, G.; Morbidi, F.; Mariottini, G.L. Hierarchical control of a team of quadrotors for cooperative active target tracking. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013.
14. Olfati-Saber, R.; Jalalkamali, P. Collaborative target tracking using distributed Kalman filtering on mobile sensor networks. In Proceedings of the 2011 American Control Conference, San Francisco, CA, USA, 29 June–1 July 2011.
15. Julian, B.J.; Angermann, M.; Schwager, M.; Rus, D. Distributed robotic sensor networks: An information-theoretic approach. *Int. J. Robot. Res.* **2012**, *31*, 1134–1154. [[CrossRef](#)]
16. Mahony, R.; Kumar, V.; Corke, P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robot. Autom. Mag.* **2012**, *19*, 20–32. [[CrossRef](#)]
17. Zhao, S.; Chen, B.M.; Lee, T.H. Optimal sensor placement for target localisation and tracking in 2D and 3D. *Int. J. Control* **2013**, *86*, 1687–1704. [[CrossRef](#)]
18. Andreetto, M.; Pacher, M.; Fontanelli, D.; Macii, D. A Cooperative Monitoring Technique Using Visually Servoed Drones. In Proceedings of the 2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS), Trento, Italy, 9–10 July 2015. [[CrossRef](#)]
19. Xiao, L.; Boyd, S.; Lall, S. A scheme for robust distributed sensor fusion based on average consensus. In Proceedings of the IPSN 2005 Fourth International Symposium on Information Processing in Sensor Networks, Boise, ID, USA, 15–15 April 2005. [[CrossRef](#)]
20. Wang, X.; Fu, M.; Zhang, H. Target Tracking in Wireless Sensor Networks Based on the Combination of KF and MLE Using Distance Measurements. *IEEE Trans. Mob. Comput.* **2012**, *11*, 567–576. [[CrossRef](#)]

21. Parrot SA. Available online: <http://www.parrot.com/usa/products/bebop-drone> (accessed on 19 September 2018).
22. Huang, H.; Hoffmann, G.; Waslander, S.; Tomlin, C. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009. [[CrossRef](#)]
23. Solà, J.; Monin, A.; Devy, M.; Lemaire, T. Undelayed initialization in bearing only SLAM. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005. [[CrossRef](#)]
24. Singer, R. Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets. *IEEE Trans. Aerosp. Electron. Syst.* **1970**, *6*, 473–483. [[CrossRef](#)]
25. Ricquebourg, Y.; Bouthemy, P. Real-Time Tracking of Moving Persons by Exploiting Spatio-Temporal Image Slices. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 797–808. [[CrossRef](#)]
26. Ariyur, K.; Fregene, K. Autonomous tracking of a ground vehicle by a UAV. In Proceedings of the 2018 American Control Conference, Seattle, WA, USA, 11–13 June 2008.
27. Asaula, R.; Fontanelli, D.; Palopoli, L. A Probabilistic Methodology for Predicting Injuries to Human Operators in Automated Production lines. In Proceedings of the 2009 IEEE Conference on Emerging Technologies and Factory Automation, Mallorca, Spain, 22–25 September 2009. [[CrossRef](#)]
28. Asaula, R.; Fontanelli, D.; Palopoli, L. Safety provisions for Human/Robot Interactions using Stochastic Discrete Abstractions. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).