Decentralized Target Tracking based on Multi-Robot Cooperative Triangulation

A. Dias[1], J. Capitan[2], L. Merino[3], J. Almeida[1], P. Lima[4] and E. Silva[1]

***Abstract*— Target tracking with bearing-only sensors is a challenging problem when the target moves dynamically in complex scenarios. Besides the partial observability of such sensors, they have limited field of views, occlusions can occur, etc. In those cases, cooperative approaches with multiple tracking robots are interesting, but the different sources of uncertain information need to be considered appropriately in order to achieve better estimates. Even though there exist probabilistic filters that can estimate the position of a target dealing with uncertainties, bearing-only measurements bring usually additional problems with initialization and data association. In this paper, we propose a multi-robot triangulation method with a dynamic baseline that can triangulate bearing-only measurements in a probabilistic manner to produce 3D observations. This method is combined with a decentralized stochastic filter and used to tackle those initialization and data association issues. The approach is validated with simulations and field experiments where a team of aerial and ground robots with cameras needs to track a dynamic target.**

I. INTRODUCTION

Over the last years, there has been an increasing research effort on multi-robot cooperative perception, to ensure robust and reliable autonomous perception in real scenarios involving dynamic environments and varying perception conditions. Tracking mobile targets with bearing-only sensors is a clear example where combining information from different robots can be essential if the targets move very dynamically in complex scenarios. In addition, cooperative 3D target estimation is useful in many applications combining static and dynamic cameras, such as search and rescue [1] and border surveillance [2].

There are stochastic filters that model uncertainties probabilistically and fuse data from sensors to estimate the position of one or several targets. Depending on the probability distribution, different representations can be used, such as Bayes Filters [3], Particle Filters [4] or Kalman/Information Filters [5]. Still, there are some issues that make the problem challenging [6], [7]: (i) sensors have different levels of accuracy that should be weighted accordingly; (ii) outliers or measurements coming from spurious data should be discarded; and (iii) data association and initialization of the estimation must be performed. This last issue is particularly relevant with bearing-only sensors, which lack depth information and absolute scale [8]. Some techniques, such as monocular vision system Structure-from-Motion or Visual Simultaneous Localization and Mapping, managed to combine bearing-only observations to estimate depth with a high accuracy, both in indoor and outdoor map-building applications [9], [10]. However, this level of accuracy presents some constraints, such as high computational requirements, and cameras with low dynamics and large fields of view.

Additionally, other works proposed solutions to cope with initialization and data association within the estimation filters [7]. Instead, we propose a solution at the level of the perception sensors, i.e., when generating the measurements that the filter will integrate. In particular, we apply this idea for cooperative 3D target tracking with multiple cameras on board mobile robots. We use a method that estimates a 3D observation of a target from the monocular vision measurements of several robots. We first introduced this method, named Uncertainty-based Multi-Robot Cooperative Triangulation (UCoT), as a standalone component [11]. However, in this work, we propose to use it as a novel multi-robot *sensor* integrated within a decentralized stochastic filter for data fusion [12].

UCoT is a triangulation method with a dynamic stereo baseline that weights different monocular observations according to their uncertainties in a probabilistic manner, and produces a single 3D measurement. Thus, instead of integrating directly into the filter the bearing-only measurements from the monocular cameras, the idea is to use UCoT to pre-process them and generate 3D measurements that will be used locally by each robot filter. Moreover, we use a Decentralized Delayed-State Information Filter (DDSIF) [12] that allows the robots to share their local information with other team-mates. Our main contributions are the following:

- Contrary to other triangulation methods with mobile targets and cameras [10], UCoT does not require features available between frames for batch recursive 3D estimation.
- UCoT allows our DDSIF to be initialized without additional assumptions. Other filters based on bearing-only observations need to make assumptions on the initial height or size of the target.
- UCoT improves the data association phase of the DDSIF, discarding outliers, by means of a probabilistic validation. Pairs of bearing-only observations whose triangulation are not good enough are considered inappropriate or very noisy.
- The approach is scalable and flexible, based on a de-

[1]A. Dias, J. Almeida and E. Silva are with INESC Technology and Science, ISEP - School of Engineering, Porto, Portugal adias,jma,eaps@lsa.isep.ipp.pt

[2]J. Capitan is with the University of Seville, Seville, Spain jcapitan@us.es

[3]L. Merino is with the Pablo de Olavide University, Seville, Spain lmercab@upo.es

[4]P. Lima is with Institute for Systems and Robotics, Instituto Superior técnico, Lisbon, Portugal pal@isr.utl.pt

centralized filter and local communication. Estimations are computed locally at each robot by exchanging information with others.

The paper is organised as follows: Section II introduces the general architecture of the proposed approach and describes the decentralized estimation filter for target tracking; Section III describes UCoT, the Uncertainty-based Multi-Robot Cooperative Triangulation; Section IV discusses the advantages of using UCoT as a virtual sensor; Section V provides experimental results to validate the method; and Section VI concludes the paper and suggests future work.

II. Decentralized Data Fusion for Target Tracking

This section describes the general architecture of the approach presented in this paper for target tracking, which is based on the decentralized Information Filter depicted in Fig. 1. The filter used (DDSIF) was previously introduced by some of the authors and its details can be found in [12]. Each robot runs a local instance of the DDSIF and computes a local belief over the state of the target based on local measurements. Then, this belief is shared with other robots and the information (beliefs) coming from others is fused with the local estimate. Thus, the filter can provide estimates in a decentralized fashion even when the robots are out of communication range. Once they get closer again, their beliefs will be fused, avoiding losses of information. The decentralized estimation converges to the one that would be obtained by a centralized filter as long as the robots communicate in a tree-like network [12].

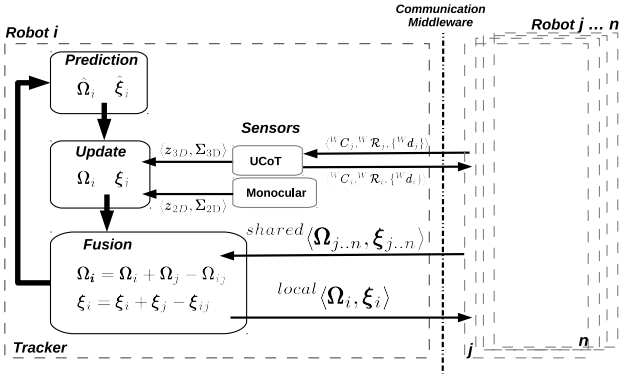

Fig. 1: Architecture of the proposed decentralized data fusion approach.

DDSIF - Decentralized Delayed-state Information Filter

Similarly to the Kalman Filter, the Information Filter assumes Gaussian probability distributions, but maintains a estimation of the information vector $\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and information matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance matrix of the estimated state respectively.

In this case, the state to estimate consists of the 3D position and velocity of a moving target. There is a step to *predict* the position and velocity of the target, and a step to *update* the belief with the local measurements. In the prediction step, the velocity is assumed to be affected by an acceleration modelled as zero-mean white noise, while the position changes according to that velocity and the step duration.

Each robot is supposed to carry a camera, and two different kinds of measurements are considered in the system: $\boldsymbol{z}_{2D}$ and $\boldsymbol{z}_{3D}$. The former consists of a position of the target on the image plane, the latter is the result of the UCoT sensor proposed in this paper and consists of a 3D position of the target in the global coordinate system. The usual pin-hole projection is used to model the 2D measurements from the target state. This is a non-linear model and a first-order linearization is applied. However, the model for the 3D measurements is straightforward. Gaussian additive noise is considered in both cases.

When there is information available from other robots, a fusion step needs to be performed. Due to the additive nature of its update step, the Information Filter allows robots to do this easily. For example, if robot i receives the belief of robot j ($\boldsymbol{\xi}_j, \boldsymbol{\Omega}_j$), it updates the local belief with the following rule:

$$\begin{aligned} \boldsymbol{\xi}_i &= \boldsymbol{\xi}_i + \boldsymbol{\xi}_j - \boldsymbol{\xi}_{ij} \\ \boldsymbol{\Omega}_i &= \boldsymbol{\Omega}_i + \boldsymbol{\Omega}_j - \boldsymbol{\Omega}_{ij}, \end{aligned} \tag{1}$$

where $\boldsymbol{\xi}_{ij}$ and $\boldsymbol{\Omega}_{ij}$ represent the information previously exchanged between robots i and j. This common information must be removed first not to get overconfident estimations. Moreover, it can be computed by a paralell filter as long as the robots communicate in a tree-like network. When this cannot be assured, other conservative fusion rules, such as the Covariance Intersection can be applied, but the decentralized estimation losses some information regarding the centralized one [12].

Finally, it is important to mention that the DDSIF maintains trajectories over the state instead of just the last state. This allows robots to integrate local measurements or beliefs from others that arrived delayed due to communication issues, and, in the linear case, to recover the same estimation as a centralized filter (with a certain lag depending on the communication hops in the network) [12].

III. UCoT - Uncertainty-based Multi-Robot Cooperative Triangulation

In this paper, the DDSIF in Section II is complemented with a *virtual* sensor that provides 3D measurements, the Uncertainty-based Multi-Robot Cooperative Triangulation (UCoT). This sensor allows the framework to integrate 3D information based on monocular 2D measurements, using the relative position and attitude provided by each robot, and based on geometric constraints derived from triangulation [13] (see Fig. 2). In addition, the uncertainties of the observation model, position and attitude of each robot, are modelled using a first-order uncertainty propagation, with the assumption that all sources of uncertainty can be modelled as uncorrelated Gaussian noises.

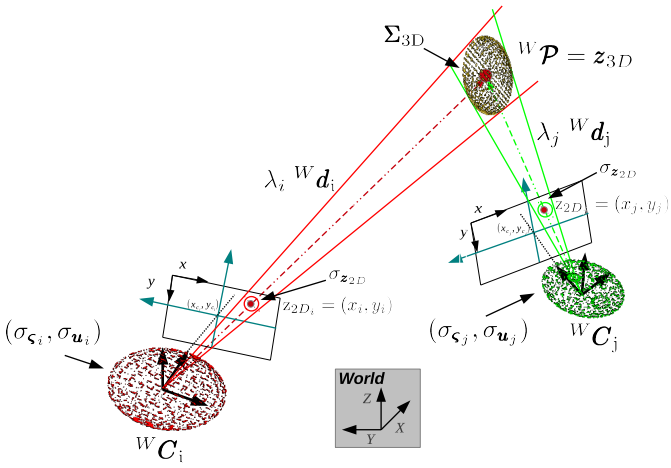

Fig. 2: Relative pose between robots i and j, each of them equipped with a monocular vision system, estimating the 3D target position.

Triangulation based on Uncertainty

Considering the formulation by Trucco [13] relative to the mid-point triangulation using a stereo rigid baseline, the following equation is proposed to use a dynamic baseline:

$$ {}^W\boldsymbol{\mathcal{P}} = \pi_i\ {}^W\boldsymbol{\mathcal{P}}_i + \pi_j\ {}^W\boldsymbol{\mathcal{P}}_j \tag{2}$$

where ${}^W\boldsymbol{\mathcal{P}}_i = {}^W\boldsymbol{C}_i + \lambda_i\ {}^W\boldsymbol{d}_i$ is the point on the line with origin ${}^W\boldsymbol{C}_i$ (the position of camera i) and unitary direction vector ${}^W\boldsymbol{d}_i$ (given by the position of the target in pixel coordinates) corresponding to a particular value of λ_i (equivalently for ${}^W\boldsymbol{\mathcal{P}}_j$).

This equation represents a dynamic baseline approach that computes a 3D position of a target in the global frame[1] ${}^W\boldsymbol{\mathcal{P}}$ by weighting the bearing-only measurements from a pair of monocular cameras defined as i, j. If ${}^W\boldsymbol{d}$ are the unitary direction vectors of the cameras' rays pointing to the target, according to the traditional stereo geometry with rigid baseline, the resulting 3D point will be located in a line perpendicular to both rays, represented by a direction vector ${}^W\boldsymbol{d}_\perp = {}^W\boldsymbol{d}_i \wedge {}^W\boldsymbol{d}_j$. If the baseline is dynamic, the cameras also need to share their global positions ${}^W\boldsymbol{C}$ and attitudes ${}^W\boldsymbol{\mathcal{R}}$, as well as their direction vectors. Applying the stereo geometry with all these data, a linear system can be solved [11] to obtain the parameters λ_i, λ_j, and hence, the 3D points corresponding to each camera, ${}^W\boldsymbol{\mathcal{P}}_i$, ${}^W\boldsymbol{\mathcal{P}}_j$. The geometric representation of the process is depicted in Fig. 2.

Once we have the line defined by ${}^W\boldsymbol{d}_\perp$, instead of selecting the mid-point between ${}^W\boldsymbol{\mathcal{P}}_i$ and ${}^W\boldsymbol{\mathcal{P}}_j$, as done in [13], we propose the weights π_i and π_j, which will be derived in Eq. (6) to consider appropriately the uncertainties of each monocular system.

The covariance matrix of the 3D target position $\boldsymbol{\Sigma}_{3D}$ is estimated considering all the sources of uncertainty. For each camera, it is assumed that there is uncertainty in the target location in pixel coordinates $\boldsymbol{\sigma}_{z_{2D}} = diag[\sigma_x, \sigma_y]$; as well as in the global camera position provided by a GPS $\boldsymbol{\sigma}_{\boldsymbol{\varsigma}} = diag[\sigma_\lambda, \sigma_\varphi, \sigma_h]$, where $\boldsymbol{\varsigma} = (\lambda, \varphi, h)$ are the latitude, longitude and altitude, respectively; and in the attitude provided by an IMU $\boldsymbol{\sigma}_{\boldsymbol{u}} = diag[\sigma_\phi, \sigma_\theta, \sigma_\psi]$, where $\boldsymbol{u} = (\phi, \theta, \psi)$ are the roll, pitch and yaw angles, respectively. All of them are modelled as uncorrelated zero-mean Gaussian random variables.

A vector with all the uncertain variables in Eq. (2) can be composed $\boldsymbol{\nu}_{(i,j)} = [\boldsymbol{\varsigma}_i, \boldsymbol{u}_i, \boldsymbol{z}_{2D_i}, \boldsymbol{\varsigma}_j, \boldsymbol{u}_j, \boldsymbol{z}_{2D_j}]$. Then, using a first-order uncertainty propagation, it is possible to approximate the uncertainty on the 3D target position as follows:

$$ \boldsymbol{\Sigma}_{3D} = \boldsymbol{J}_{\boldsymbol{\mathcal{P}}} \boldsymbol{\Lambda}_{i,j} \boldsymbol{J}_{\boldsymbol{\mathcal{P}}}{}^T, \tag{3}$$

where $\boldsymbol{J}_{\boldsymbol{\mathcal{P}}}$ stands for the Jacobian matrix of ${}^W\boldsymbol{\mathcal{P}}$ in Eq. (2) with respect to the noisy variables

$$ \boldsymbol{J}_{\boldsymbol{\mathcal{P}}[3\times16]} = \nabla_{\boldsymbol{\nu}_{(i,j)}}\ {}^W\boldsymbol{\mathcal{P}}(\boldsymbol{\nu}_{(i,j)}), \tag{4}$$

and $\boldsymbol{\Lambda}_{i,j}$ is the input covariance matrix represented by a diagonal line relative to all sources of uncertainty for both cameras $(\boldsymbol{\nu}_{(i,j)})$.

In order to ensure that all sources of uncertainty from each intersection ray are addressed in a probabilistic manner when obtaining the weights associated with each ray, once again it is necessary to estimate the covariances $\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}i}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}j}$ using a first-order propagation:

$$\begin{array}{ll} \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}i} = \boldsymbol{J}_{\boldsymbol{\mathcal{P}}i} \boldsymbol{\Lambda}_{i,j} \boldsymbol{J}_{\boldsymbol{\mathcal{P}}i}^T & \boldsymbol{J}_{\boldsymbol{\mathcal{P}}i[3\times16]} = \nabla_{(\boldsymbol{\nu}_{(i,j)})}\ {}^W\boldsymbol{\mathcal{P}}_i \\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}j} = \boldsymbol{J}_{\boldsymbol{\mathcal{P}}j} \boldsymbol{\Lambda}_{i,j} \boldsymbol{J}_{\boldsymbol{\mathcal{P}}j}^T & \boldsymbol{J}_{\boldsymbol{\mathcal{P}}j[3\times16]} = \nabla_{(\boldsymbol{\nu}_{(i,j)})}\ {}^W\boldsymbol{\mathcal{P}}_j \end{array} \tag{5}$$

where $\boldsymbol{J}_{\boldsymbol{\mathcal{P}}i}$ and $\boldsymbol{J}_{\boldsymbol{\mathcal{P}}j}$ are the Jacobian matrices from ${}^W\boldsymbol{\mathcal{P}}_i$ and ${}^W\boldsymbol{\mathcal{P}}_j$, respectively. Therefore, with the uncertainties $\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}i}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}j}$, each camera contribution can be weighted accordingly in the line described by the perpendicular vector ${}^W\boldsymbol{d}_\perp$:

$$\begin{array}{l} \pi_i = \frac{({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}j}\ {}^W\boldsymbol{d}_\perp{}^T)^2}{({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}i}\ {}^W\boldsymbol{d}_\perp{}^T)^2 + ({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}j}\ {}^W\boldsymbol{d}_\perp{}^T)^2} \\ \pi_j = \frac{({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}i}\ {}^W\boldsymbol{d}_\perp{}^T)^2}{({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}i}\ {}^W\boldsymbol{d}_\perp{}^T)^2 + ({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}j}\ {}^W\boldsymbol{d}_\perp{}^T)^2} \end{array} \tag{6}$$

Combining the probabilistic weights π_i, π_j from Eq. (6) and the dynamic baseline triangulation in Eq. (2), it is possible to obtain the 3D target estimation $(\boldsymbol{z}_{3D})$ provided by UCoT:

$$\begin{array}{l} {}^W\boldsymbol{\mathcal{P}} = \frac{({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}j}\ {}^W\boldsymbol{d}_\perp{}^T)^2}{({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}i}\ {}^W\boldsymbol{d}_\perp{}^T)^2 + ({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}j}\ {}^W\boldsymbol{d}_\perp{}^T)^2} ({}^W\boldsymbol{C}_i + \lambda_i^W\boldsymbol{d}_i) \\ \quad + \frac{({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}i}\ {}^W\boldsymbol{d}_\perp{}^T)^2}{({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}i}\ {}^W\boldsymbol{d}_\perp{}^T)^2 + ({}^W\boldsymbol{d}_\perp\ \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{P}}j}\ {}^W\boldsymbol{d}_\perp{}^T)^2} ({}^W\boldsymbol{C}_j + \lambda_j^W\boldsymbol{d}_j) \end{array} \tag{7}$$

Multi-robot Features Data Association

The above formulation is used to combine a pair of bearing-only measurements from two cameras. However, not all pairs are considered valid. First, the normalized squared innovation for the 3D intersection between the two monocular observations is computed:

$$ ({}^W\boldsymbol{\mathcal{P}}_i - {}^W\boldsymbol{\mathcal{P}}_j)^T \boldsymbol{\Sigma}_{3D}^{-1} ({}^W\boldsymbol{\mathcal{P}}_i - {}^W\boldsymbol{\mathcal{P}}_j) < \varepsilon_{3D}, \tag{8}$$

[1]Throughout the paper, the super-index W indicates that the variable is expressed in the global coordinate system.

where ε_{3D} follows a chi-square distribution. Only pairs with an innovation good enough, i.e., those fulfilling Eq. (8), are considered valid. The gate's bounding values to ensure a valid pair can be obtained from a cumulative χ^2 table with 3 degrees of freedom. This method allows the UCoT sensor to detect spurious observations (pairs not matching) and discard them as outliers. Moreover, the method can be used for data association in case of multiple targets in the scenario. If there are observations at the same instant from more than two cameras, UCoT selects the pair with a better innovation and provides 3D point corresponding to that pair.

IV. Discussion

The DDSIF presented in Section II allows a team of robots with on-board cameras to estimate the position of a target in a decentralized fashion. The filter can integrate 2D bearing-only measurements from the cameras as usual, but it also includes the novel possibility of using the 3D measurements computed by the UCoT method explained in Section III.

As it was shown in Fig 1, both the DDSIF and the UCoT allow the robots to share information. Robots share their local estimates on the target 3D position and fuse beliefs from others thanks to the DDSIF. Besides, they share their camera position and attitude, as well as their direction vectors pointing to the target. This information is used by UCoT to generate 3D measurements of the target position and for data association. This flexibility allows the system to update information just locally when needed, or to fuse information from others when is available.

Additionally, the generation of 3D measurements by UCoT can help the DDSIF by initializing the height estimate. In a simpler filter only integrating bearing-only measurements, the initial position of the target could be computed by projecting the first measurement into the ground. However, we need to assume that the target will stay on the ground, or more generally, that its initial height is known. For example, this assumption is very restrictive in the case of aerial targets [14].

Another improvement derived by the inclusion of the UCoT sensor into the filter is the phase of data association. Thanks to the methods explained in Section III, a robot running UCoT locally can evaluate the appropriateness of bearing-only measurements received from other cameras. If those measurements do not fit probabilistically with the local target estimate, UCoT can consider them as outliers and discard them. This method allow us to eliminate spurious measurements, and helps the filter to converge and reduce the noise in the final estimate.

V. EXPERIMENTS

This section describes the experimental results to assess the impact of integrating UCoT as a virtual sensor with a DDSIF. Two outdoor experimental cases are proposed: a simulated scenario where two Micro Aerial Vehicles (MAV) are tracking a moving target; and a real scenario where an MAV and an Unmanned Ground Vehicle (UGV) are the trackers. In both cases, the terrain is not totally flat, but presents an altitude variation of around 2 meters in the simulation and 7 meters in the field experiments. The ground truth of the target position is available thanks to a high accuracy RTK-GPS sensor with an error lower than $10cm$. In both cases, an image processing component is run on each robot to detect the target on the image plane. The target has a distinctive color and the algorithm is based on blob detection.

A. Simulations

The simulated environment was created [17] with the realistic robotic simulator MORSE [2] and is depicted in Fig. 3. The target is simulated with another ground or aerial vehicle, depending on the experiment. Moreover, it follows the same fixed path during the experiments, which is unknown for the trackers.

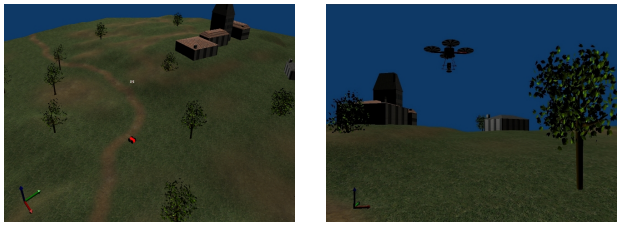

Fig. 3: **Left**: Simulated environment in MORSE. **Right**: One of the MAV trackers with a camera pointing downwards.

In order to analyse different circumstances during the simulations, all the configurations shown in Fig. 4 were tested. In configurations (a), (b), and (c), the target was simulated by a UGV that was moving on the ground, so its variation in terms of altitude was small. Moreover, the trackers were following the target in different geometric formations to test their effect on the final estimate. In configuration (d), the target was simulated by another MAV that was also varying its altitude, which can show how the system performs with changes in that component.

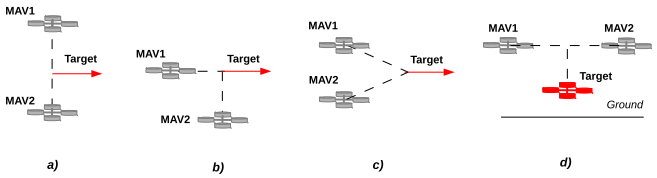

Fig. 4: Different spatial configurations for two MAVs tracking a moving target.

Table I shows average results for each configuration comparing several approaches. In the first three approaches the DDSIF is run on each robot with the fusion rule as depicted in Fig. 1, while in the last one a conservative fusion rule is used (Covariance Intersection). In the method **2D-2D**, both trackers are integrating only 2D measurements; in **2D-3D** trackers integrate 3D measurements from UCoT or 2D measurements when there is no 3D available (no 2D measurements from the two cameras at that instant); in

[2]http://www.openrobots.org/wiki/morse

Configuration	2D - 2D				2D - 3D				3D (UCoT)				3D (UCoT-CI)			
	X	Y	Z	ϵ_{NEES}	X	Y	Z	ϵ_{NEES}	X	Y	Z	ϵ_{NEES}	X	Y	Z	ϵ_{NEES}
a)	0.090	0.441	2.627	46.698	0.068	0.642	0.321	0.977	0.084	0.149	0.334	0.776	0.087	0.166	0.350	0.808
b)	0.482	1.482	11.520	34.818	0.066	0.201	1.262	1.206	0.054	0.154	0.861	1.404	0.165	0.179	0.173	0.915
c)	1.210	2.942	14.482	42.318	0.195	0.153	0.223	1.006	0.112	0.086	0.162	1.025	0.081	0.296	1.332	1.238
d)	0.057	0.928	3.334	51.597	0.039	0.768	0.340	2.861	0.018	0.035	0.251	1.090	0.027	0.045	0.332	0.831

TABLE I: RMS error on the 3D target estimate (meters) and ϵ_{NEES} for each spatial configuration. The DDSIF is run in two MAV trackers with different fusion approaches. For simplicity, only the results in one of the trackers are shown.

3D (UCoT) and **3D (UCoT-CI)** only measurements from UCoT are integrated, but Covariance Intersection is used to fuse beliefs in the latter. Moreover, during the experiments, spurious and noisy observations from the cameras were simulated to see the system performance.

The Root Mean Square (RMS) error of the estimates with respect to the ground truth are shown, as well as the Normalized Estimation Error Square (ϵ_{NEES}). The second metric is useful to evaluate the consistency of the filter estimate with respect to the actual value [15], [16]. The ϵ_{NEES} can be compared with a χ^2 distribution with three degrees of freedom in order to assess whether the filter tends to be pessimistic or overestimate its capabilities. It can be seen that the introduction of the 3D observations provided by UCoT improves the estimate error, mainly in the Z component. Also, the consistency is improved, since ϵ_{NEES} values are lower. With the method **3D (UCoT)**, the spurious measurements were discarded by UCoT. However, with the method **2D-3D**, some 2D measurements, which may be noisy, were still included. This is why the results of **2D-3D** are slightly worse. Note that results between **3D (UCoT)** and **3D (UCoT-CI)** are pretty similar. Even though some information is lost with the Covariance Intersection, the filter still achieves a good performance.

Due to space limitations, plots in Fig. 5 and Fig. 6 show the full trajectories for the simulations only for configurations a) and d) respectively, which are more relevant. In the case of the method **2D-2D**, in order to converge, the filter was initialized assuming the target height as known. In the other cases, the filter was initialized with the 3D measurements from UCoT. Nonetheless, the **2D-2D** method presents a peak in ϵ_{NEES} at the beginning due to the worse initialization. In Fig. 5, the impact of the outliers can be seen at instants $260s$ and $320s$, where the estimate starts to diverge for the **2D-2D** case. A peak in ϵ_{NEES} can also be observed at the same instants. The impact of the outliers with the method **2D-3D** is only observed at instant $260s$, where the filter was without UCoT measurements, and hence, without data association. However, this effect is not present at instant $320s$, which means that 3D measurements from UCoT were available. With the method **3D (UCoT)**, the outliers were rejected thanks to the data association in Eq. (8). Similar results are depicted in Fig. 6, at the instant $60s$. The **3D (UCoT-CI)** gives similar results and is not presented here due to space limitations.

B. Field Experiments

Field experiments were also performed to prove the feasibility of the system in a non-urban area with several landscape elements including vegetation and rocks, as depicted in Fig. 7[3].

Fig. 7: Experimental scenario with the UGV and the MAV tracking a person. Images from the UGV and MAV cameras.

The tracker robots used were a UGV [18] and the Asctec Pelican MAV. The UGV carries two cameras with a resolution of 1278×958 in a stereo rigid baseline (~ 0.76 meters), a Novatel GPS receiver and an IMU Microstrain. The Pelican MAV is a commercial platform to which a downward monocular camera with a resolution of 1280×1024 was added. The target consisted of a person moving along the environment at the velocity of $\sim 0.8m/s$. The person was wearing distinctive color clothes in order to help the image processing algorithms on the cameras.

Both robots were tracking the target, the MAV hovered over it (~ 20 meters), and the UGV performed an approximation manoeuvre with a safe distance of ~ 2 meters. However, most of the time the UGV is at a distance between 5 to 10 meters, and the MAV relative height is between 5 to 20 meters, due to the ground gradient of the scenario where the target is moving.

The UGV is equipped with a fixed stereo rigid baseline able to estimate the 3D position of the target, although the accuracy depends on the distance to the target(initially $\sim 35m$) and the available stereo baseline. As we proved in [11], the accuracy of the this stereo estimation is low for these distances, and therefore we propose to evaluate the DDSIF by combining the MAV 2D measurements, and the 3D measurements obtained by UCoT with the monocular information the MAV camera and one of the UGV cameras. The results are detailed in Table II and depicted in Fig. 8.

The overall performance is similar to the one obtained using the simulated environment. In the case of the method **2D-2D**, the filter was initialized with the known height of the person and therefore the ϵ_{NEES} peak is low, meaning that the value was coherent with the real height. The lower accuracy shown in all methods is due to the fact of being in an outdoor environment affected by light variations, and also from the robots' GPS high uncertainty $(\sim 3m)$.

[3]A video of the experiments is available at https://www.youtube.com/watch?v=OkoNYua5A9Y.

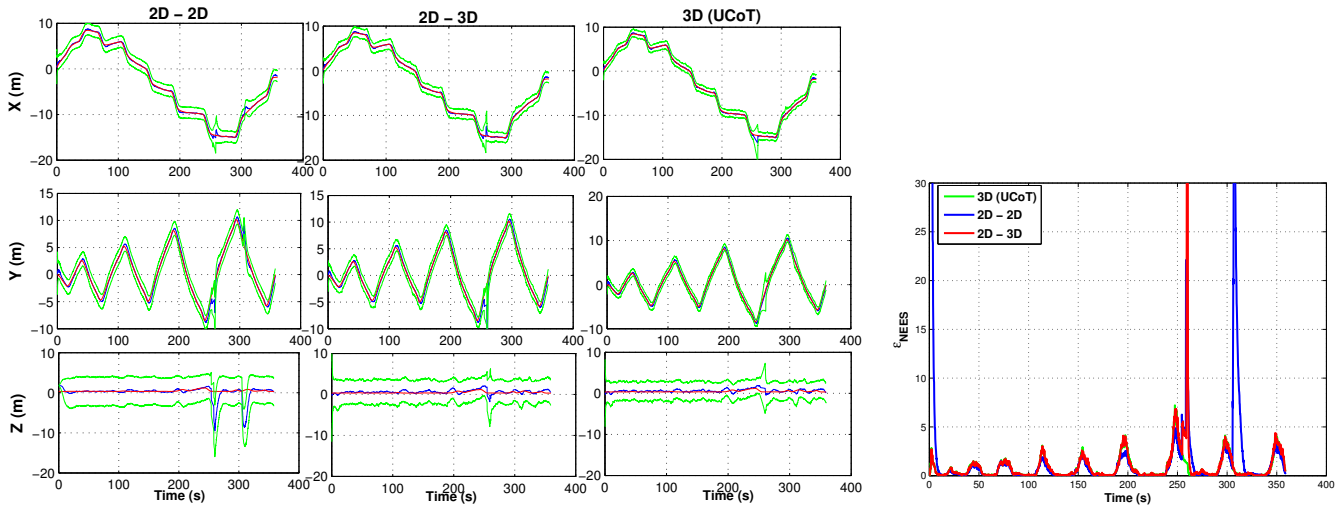

Fig. 5: Results corresponding to the simulation with the spatial configuration a). **Left**: 3D target estimate (blue) of one of the trackers for several DDSIF methods. The ground truth (red) and the confidence intervals (green) are also plotted. **Right**: ϵ_{NEES} of one of the trackers for several DDSIF methods.

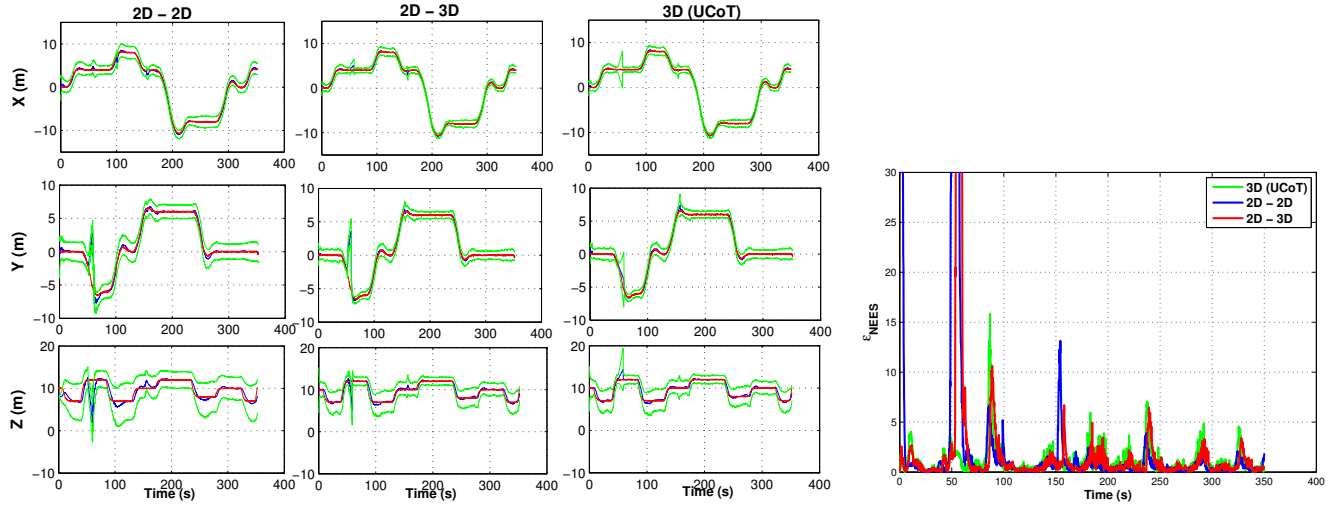

Fig. 6: Results corresponding to the simulation with the spatial configuration d). **Left**: 3D target estimate (blue) of one of the trackers for several DDSIF methods. The ground truth (red) and the confidence intervals (green) are also plotted. **Right**: ϵ_{NEES} of one of the trackers for several DDSIF methods.

Method	2D-2D	2D - 3D	3D (UCoT)
X	1.714	0.685	1.146
Y	3.945	1.737	1.249
Z	0.116	0.660	0.555
ϵ_{NEES}	77.667	49.166	17.142

TABLE II: Field experiment. RMS error on the 3D target estimate (meters) and ϵ_{NEES} for one of the trackers for several DDSIF configurations.

VI. CONCLUSIONS

This paper proposes a multi-robot triangulation method as a novel sensor to be combined with a decentralized stochastic filter. The method was evaluated with simulations and field experiments where a team of aerial and ground robots with cameras performs a task of tracking a dynamic target.

The results from the simulations and the field experiments show how the multi-robot triangulation allows to ensure a correct filter initialization; furthermore, the method improves the data association phase, discarding outliers, by means of a probabilistic geometric validation. This leads to an improved consistency of the filter. Finally, it is shown how this can be integrated into a decentralized filter for cooperative tracking.

As future work, we intend to perform more extensively field experiments with an improvement in the robot position accuracy, by means of a Real-Time Kinematic GPS related to a ground station. Another line of work will be focused on active perception. The UCoT method is able to provide

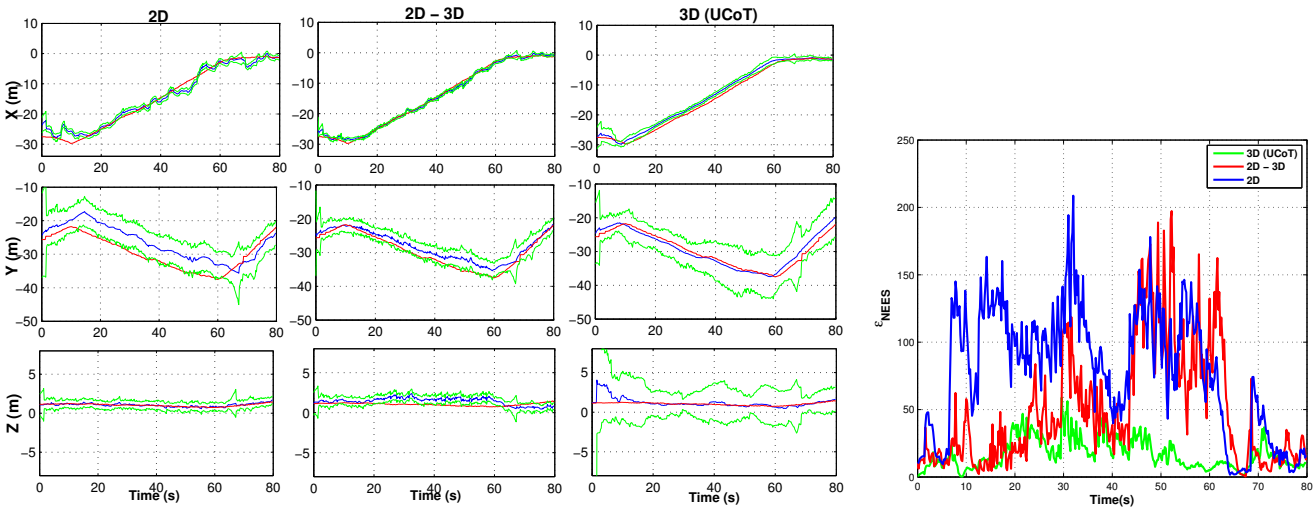

Fig. 8: Results corresponding to the field experiments. **Left**: 3D target estimate (blue) of one of the trackers for several DDSIF methods. The ground truth (red) and the confidence intervals (green) are also plotted. **Right**: ϵ_{NEES} of one of the trackers for several DDSIF methods.

the 3D estimated covariance between triangulated cameras, and therefore, by changing the geometry between them, we could reduce the uncertainty on the estimation and improve the global perception of the fleet.

ACKNOWLEDGMENTS

This work is funded by the ERDF European Regional Development Fund through the COMPETE Programme; by the FCT Portuguese Foundation for Science and Technology through the projects FCOMP-01-0124-FEDER-037281 and ISR/LARSYS PEst-OE/EEI/LA0009/2013; and by the Junta de Andalucia through the project PAIS-MultiRobot (TIC-7390).

REFERENCES

[1] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012. [Online]. Available: http://dx.doi.org/10.1002/rob.21436

[2] Z. Xu, B. Douillard, P. Morton, and V. Vlaskine, "Towards Collaborative Multi-MAV-UGV Teams for Target Tracking," in *2012 Robotics: Science and Systems workshop "Integration of perception with control and navigation for resource-limited, highly dynamic, autonomous systems"*, 2012. [Online]. Available: http://rss2012_workshop.visual-navigation.com/files/Zu.pdf

[3] E.-M. Wong, F. Bourgault, and T. Furukawa, "Multi-vehicle Bayesian search for multiple lost targets," 2005, pp. 3169–3174.

[4] L. Ong, B. Upcroft, T. Bailey, M. Ridley, S. Sukkarieh, and H. Durrant-Whyte, "A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles," in *IROS*, 2006, pp. 4539–4544.

[5] F. Morbidi and G. L. Mariottini, "On active target tracking and cooperative localization for multiple aerial vehicles," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 2229–2234.

[6] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253511000558

[7] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 12, pp. 1696–1710, Dec 2006.

[8] M. W. Achtelik, S. Weiss, M. Chli, F. Dellaert, R. Siegwart, and Z. Eth, "Collaborative Stereo," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2011.

[9] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, 2007, pp. 225–234.

[10] S. M. Weiss, "Vision based navigation for micro helicopters," Ph.D. dissertation, 2012. [Online]. Available: http://e-collection.library.ethz.ch/view/eth:5889

[11] A. Dias, J. Almeida, E. Silva, and P. Lima, "Uncertainty based multi-robot cooperative triangulation," *RoboCup Symposium 2014, Springer-Verlag Lecture Notes in Artificial Intelligence (LNAI)*, 2014.

[12] J. Capitan, L. Merino, F. Caballero, and A. Ollero, "Decentralized delayed-state information filter (DDSIF): A new approach for cooperative decentralized tracking," *Robotics and Autonomous Systems*, vol. 59, pp. 376–388, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2011.02.001

[13] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[14] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, 2006.

[15] R. Martinez-Cantin and J. A. Castellanos, "Unscented slam for large-scale outdoor environments," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3427–3432.

[16] C. Otto, *Fusion of data from heterogeneous sensors with distributed fields of view and situation evaluation for advanced driver assistance systems*. Karlsruhe: KIT Scientific Publishing, 2013. [Online]. Available: http://digbib.ubka.uni-karlsruhe.de/volltexte/1000035932

[17] A. Dias, J. Almeida, N. Dias, P. Lima, and E. Silva, "Simulation environment for multi-robot cooperative 3d target perception," in *Simulation, Modeling, and Programming for Autonomous Robots*, ser. Lecture Notes in Computer Science, D. Brugali, J. Broenink, T. Kroeger, and B. MacDonald, Eds. Springer International Publishing, 2014, vol. 8810, pp. 98–109. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-11900-7

[18] A. Martins, G. Amaral, A. Dias, C. Almeida, J. Almeida, and E. Silva, "Tigre - an autonomous ground robot for outdoor exploration," in *Autonomous Robot Systems (Robotica), 2013 13th International Conference on*, 2013, pp. 1–6.