Collaborative localization and formation flying using distributed stereo-vision

Nathan Piasco, Julien Marzat, Martial Sanfourche

Abstract— This paper considers collaborative stereo-vision as a mean of localization for a fleet of micro-air vehicles (MAV) equipped with monocular cameras, inertial measurement units and sonar sensors. A sensor fusion scheme using an extended Kalman filter is designed to estimate the positions and orientations of all the vehicles from these distributed measurements. The estimation is completed by a formation control to maximize the overlapping fields of view of the vehicles. Experimental tests for the complete perception and control loop have been performed on multiple MAVs with centralized processing on a ROS ground station.

Index Terms— collaborative localization, formation control, micro-air vehicles, stereo-vision

I. INTRODUCTION

Vision-based localization is a popular approach in the field of robotics, particularly when the robots considered are flying vehicles in GPS-denied cluttered environments. Numerous SLAM and visual odometry methods have been developed to localize a single aerial vehicle, based on either monocular data [1], [2], which suffers from depth uncertainty, or stereo-vision [3], [4], which uses additional 3D information.

Recent research has focused on fleets of MAVs, whose main interests are to carry complementary sensors on cheaper vehicles as well as their ability to cover more field [5]. The context considered is to localize all MAVs in a fleet by fusing their distributed embedded measurements. A first approach of collaborative localization consists in merging individual maps created by mono-vehicle monocular SLAMs. The algorithms from [6]–[8] propose efficient fusion strategies, however this approach suffers from the inherent drawbacks of monocular SLAMs, i.e. depth uncertainty and drift. As an alternative approach, the fusion of collaborative stereo-vision (with a varying baseline, as each vehicle moves) with IMU data has been investigated in [9] for estimating the relative pose of two MAVs. In [10], the relative localization of multiple MAVs was obtained by combining IMU measurements and an homography estimation. The same authors also proposed a formation control to maintain the fleet in a desired layout. These papers contain promising results on stereo-vision for fleet localization, however the entire estimation process was not tested in the experiments reported since the vision algorithms were emulated by motion tracking data.

The present work proposes a filtering scheme (Section II) to localize in a global frame all the MAVs of a fleet, using

N. Piasco is a Research Engineer at A.I.Mergence, F-75013 Paris, France nathan.piasco@ai-mergence.com

J. Marzat and M. Sanfourche are Research Scientists at ONERA – The French Aerospace Lab, F-91123 Palaiseau, France, julien.marzat@onera.fr, martial.sanfourche@onera.fr

their monocular cameras in a collaborative stereo-vision process and IMUs but also additional altitude measurements and linear velocity estimates. A formation control law is also proposed (Section III) for maximizing the overlap of the MAV fields of view, in order to enhance cooperative localization. The complete vision and control loop has been flight-tested on multiple Parrot® AR Drones [11] with centralized processing on a ROS ground station.

II. COLLABORATIVE LOCALIZATION

A. Notations

A vector $^r\boldsymbol{x}_i$ stands for a variable associated with vehicle i and expressed in the coordinate frame linked to vehicle r. The world inertial frame is denoted by w. Rotations are represented by quaternions according to *Hamilton* notation:

$$\overline{q} = \left[\begin{array}{c} q \\ \boldsymbol{q} \end{array} \right] \tag{1}$$

where q stands for the real part of the quaternion and the vector $\boldsymbol{q}$ is associated with the imaginary part. *Hamilton* product of two quaternions is defined as follows [12]:

$$\overline{q}_1 \otimes \overline{q}_2 = \left[\begin{array}{c} q_1 q_2 - \boldsymbol{q_1} \cdot \boldsymbol{q_2} \\ q_1 \boldsymbol{q_2} + q_2 \boldsymbol{q_1} + \boldsymbol{q_1} \times \boldsymbol{q_2} \end{array} \right] \tag{2}$$

where $\cdot$ refers to the dot product and $\times$ the cross product. Measurements and variables symbols used afterward are:

$\boldsymbol{p}$ for a position,
$\boldsymbol{\omega}$ for an angular velocity,
$\boldsymbol{v}$ for a linear velocity,
$\boldsymbol{R}(\overline{q})$ for a rotation matrix associated with $\overline{q}$.

B. MAV characteristics and sensors

Multiple Parrot® AR Drones 2.0 have been used to test the filter and the control law described in this paper. These low-cost vehicles are equipped with:

- two monocular cameras: one at the front of the MAV and the other downward,
- an IMU (3-axis accelerometers and 3-axis gyrometers),
- a sonar sensor directed downward,
- an on-board non-programmable CPU.

The communication between the MAVs and a laptop is established through a Wi-Fi connection. The MAV driver provides a pre-filtered information about linear velocity of the MAV, computed by sensor fusion between optical flow acquired by the bottom camera and IMU accelerations [11]. The sonar sensor provides an altitude measurement in the MAV frame, $^i z_i$. The MAV altitude in world coordinates can be obtained by projecting this value as $^w z_i = \boldsymbol{R}(^w\overline{q}_i)^i z_i$.

The following measurements are thus available as inputs for each vehicle in the filtering scheme:

- angular velocity $^i\boldsymbol{\omega}_i$,
- linear velocity $^i\boldsymbol{v}_i$,
- ground altitude $^w z_i$,
- raw front image (640x360 resolution).

The state vector (pose) of the i-th MAV is

$$\boldsymbol{x}_i = \begin{pmatrix} ^w\overline{q}_i \\ ^w\boldsymbol{p}_i \end{pmatrix} \qquad (3)$$

and its dynamics are modeled by

$$\begin{array}{l} ^w\dot{\overline{q}}_i = \frac{1}{2}\left(^w\overline{q}_i \otimes {}^i\overline{\omega}_i\right) \\ ^w\dot{\boldsymbol{p}}_i = \boldsymbol{R}(^w\overline{q}_i)\boldsymbol{v}_i \end{array}, \qquad (4)$$

where $^i\overline{\omega}_i = \left(0 \ ^i\boldsymbol{\omega}_i^T\right)^T$. This model assumes that faster loops regulate the orientation and velocity such that direct velocity control inputs can be applied (which is allowed by the ardrone_autonomy ROS driver[1]).

C. Distributed stereo-vision

The aim of visual pose recognition is to compute the relative pose between two cameras with known intrinsic parameters and overlapping fields of view. The estimated pose contains:

- the relative rotation from camera 2 to camera 1

$$^1\overline{q}_2{}^v \equiv {}^1\overline{q}_2, \qquad (5)$$

- the relative position, up to a scale factor λ, of camera 2 to camera 1: $^1\boldsymbol{p}_2{}^v \equiv \lambda^1\boldsymbol{p}_2$. To deal with this scale factor in the filtering scheme, the estimated translation is normalized such that

$$^1\boldsymbol{p}_2{}^v = {}^1\boldsymbol{p}_2\|^1\boldsymbol{p}_2\|^{-1}. \qquad (6)$$

The usual approach involves to first establish feature correspondences between the two images (features f_i from image 1 and f_i' from image 2 correspond to 3D points p_i) and then solve the pose recognition problem.

1) Matching method: The matching process can be divided in two steps: an extraction of features and description on each image, followed by descriptor pairing. The SIFT [13] feature points extractor and descriptor was used (OpenCV implementation). This widely-used algorithm is versatile and scale-invariant, which is important in the case of large baseline stereo-vision problems.

2) Relative pose computation: Once the feature correspondences have been established, the five-point algorithm [14] is used to recover the relative pose between the two cameras (OpenGV [15] implementation). Nistér's algorithm computes the essential matrix $\boldsymbol{E}$ that contains the relative rotation and translation information $\boldsymbol{E} = \boldsymbol{R}[\boldsymbol{t}]_\times$, where $[\boldsymbol{t}]_\times$ denotes a skew-symmetric matrix of $\boldsymbol{t}$,

$$[\boldsymbol{t}]_\times = \begin{pmatrix} 0 & -t_x & t_y \\ t_z & 0 & -t_z \\ -t_y & t_x & 0 \end{pmatrix}. \qquad (7)$$

An *SVD* decomposition of $\boldsymbol{E}$ makes it possible to recover the relative rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$. As the matching part provides most of the time more than five corresponding features, a RANSAC step was introduced to minimize the re-projection error of the feature points on the essential matrix [15]. In the experiments, the computation time of the pose reconstruction process (synchronized images) was always lower than 0.5s on a laptop with an Intel® i7 processor, while the filter frequency was set to 15Hz.

D. Filtering scheme

This section describes the sensor fusion scheme, which consists in an Extended Kalman Filter (EKF), to estimate positions and orientations of all the MAVs.

1) Two-vehicle case: Before considering a complete fleet of MAVs, the filter design is presented for estimating poses of two vehicles in the world frame. The state vector considered contains the poses of the two MAVS as

$$\boldsymbol{X} = \left(^w\overline{q}_1^T \ ^w\overline{q}_2^T \ ^w\boldsymbol{p}_1^T \ ^w\boldsymbol{p}_2^T\right)^T. \qquad (8)$$

Unlike [9], the scale factor λ resulting from the stereo-vision process is not needed here, due to the normalization from equation (6). The input vector is composed of the variables collected from the IMUs and embedded velocity filters:

$$\boldsymbol{U} = \left(^1\overline{\omega}_1^T \ ^2\overline{\omega}_2^T \ ^1\boldsymbol{v}_1^T \ ^2\boldsymbol{v}_2^T\right)^T. \qquad (9)$$

The measurement vector incorporates the relative pose computed by the stereo-vision process and the absolute ground altitudes measured by the sonar sensors:

$$\boldsymbol{Y} = \left((^1\boldsymbol{p}_2{}^v)^T \ (^1\overline{q}_2{}^v)^T \ ^w z_1 \ ^w z_2\right)^T. \qquad (10)$$

It is assumed that the distance from the camera to the IMU is small enough to be negligible. Deriving the state vector leads to the following formulation:

$$\begin{array}{rcl} \dot{\boldsymbol{X}} &=& f(\boldsymbol{X}, \boldsymbol{U} + \boldsymbol{W}) \\ \\ \dot{\boldsymbol{X}} &=& \begin{pmatrix} ^w\dot{\overline{q}}_1 \\ ^w\dot{\overline{q}}_2 \\ ^w\dot{\boldsymbol{p}}_1 \\ ^w\dot{\boldsymbol{p}}_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\left(^w\overline{q}_1 \otimes ({}^1\overline{\omega}_1 + \boldsymbol{\eta}_{\overline{\omega}})\right) \\ \frac{1}{2}\left(^w\overline{q}_2 \otimes ({}^2\overline{\omega}_2 + \boldsymbol{\eta}_{\overline{\omega}})\right) \\ \boldsymbol{R}(^w\overline{q}_1)(^1\boldsymbol{v}_1 + \boldsymbol{\eta}_{\boldsymbol{v}}) \\ \boldsymbol{R}(^w\overline{q}_2)(^2\boldsymbol{v}_2 + \boldsymbol{\eta}_{\boldsymbol{v}}) \end{pmatrix} \end{array} (11)$$

where $\boldsymbol{W} = (\boldsymbol{\eta}_{\overline{\omega}}^T \ \boldsymbol{\eta}_{\overline{\omega}}^T \ \boldsymbol{\eta}_{\boldsymbol{v}}^T \ \boldsymbol{\eta}_{\boldsymbol{v}}^T)^T$ is the zero-mean Gaussian input noise vector, whose variances are reported on the diagonal of the input covariance matrix $\boldsymbol{Q}$. The measurement vector can be expressed according to the state variables as

$$\begin{array}{rcl} \boldsymbol{Y} &=& h(\boldsymbol{X}) + \boldsymbol{V} \qquad\qquad\qquad (12) \\ \\ &=& \begin{pmatrix} \boldsymbol{R}(^w\overline{q}_1)\left(^w\boldsymbol{p}_1 - {}^w\boldsymbol{p}_2\right)\lambda_{12} \\ ^w\overline{q}_1^* \otimes {}^w\overline{q}_2 \\ ^w p_{1,z} \\ ^w p_{2,z} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\eta}_{\boldsymbol{p}_r} \\ \boldsymbol{\eta}_{\boldsymbol{q}_r} \\ \eta_z \\ \eta_z \end{pmatrix} \end{array}$$

where:

- $\boldsymbol{V} = \left(\boldsymbol{\eta}_{\boldsymbol{p}_r}^T \ \boldsymbol{\eta}_{\boldsymbol{q}_r}^T \ \eta_z \ \eta_z\right)^T$ is the zero-mean Gaussian measurement noise vector, whose variances are placed on the diagonal of the output covariance matrix $\boldsymbol{R}$,

- $^wp_{i,z}$ stands for the third component of vector ^wp_i,
- $^w\bar{q}_1^*$ represents conjugate quaternion of $^w\bar{q}_1$ [9],
- the scale factor is taken equal to the inverse of the distance between the MAVs, $\lambda_{ij} = \|^wp_i - {}^wp_j\|^{-1}$. The predicted positions are used and considered as constant in the filter derivation. This approximation performed better in preliminary tests than introducing the exact linearization, which showed numerical stability issues.

The Euler discretization of equation (11) yields

$$
\begin{aligned}
\boldsymbol{X}_{k+1/k} &= \boldsymbol{X}_{k/k} + f(\boldsymbol{X}_{k/k}, \boldsymbol{U}_k)\Delta t \\
&= \tilde{f}(\boldsymbol{X}_{k/k}, \boldsymbol{U}_k).
\end{aligned}
\tag{13}
$$

Kalman equations give the evolution of the matrix covariance error attached to the state vector,

$$
\boldsymbol{P}_{k+1/k} = \boldsymbol{F}_k \boldsymbol{P}_{k/k} \boldsymbol{F}_k^T + \boldsymbol{L}_k \boldsymbol{Q} \boldsymbol{L}_k^T
\tag{14}
$$

where $\boldsymbol{F}_k = \left.\dfrac{\partial \tilde{f}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{X}_{k/k}, \boldsymbol{U}_k}$, $\boldsymbol{L}_k = \left.\dfrac{\partial \tilde{f}}{\partial \boldsymbol{U}}\right|_{\boldsymbol{X}_{k/k}, \boldsymbol{U}_k}$. (15)

The state vector is updated with the $\boldsymbol{Y}_k$ vector:

$$
\begin{aligned}
\boldsymbol{K}_f &= \boldsymbol{P}_{k+1/k} \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_{k+1/k} \boldsymbol{H}_k^T + \boldsymbol{R})^{-1} \\
\boldsymbol{X}_{k+1/k+1} &= \boldsymbol{X}_{k+1/k} + \boldsymbol{K}_f\left(\boldsymbol{Y}_k - h(\boldsymbol{X}_{k+1/k})\right) \\
\boldsymbol{P}_{k+1/k+1} &= (\boldsymbol{I} - \boldsymbol{K}_f \boldsymbol{H}_k)\boldsymbol{P}_{k+1/k}
\end{aligned}
\tag{16}
$$

where $\boldsymbol{K}_f$ is the Kalman gain and $\boldsymbol{H}_k = \left.\dfrac{\partial h}{\partial \boldsymbol{X}}\right|_{\boldsymbol{X}_{k/k}}$.

2) General case: The EKF can be generalized for a fleet of N MAVs. The state vector becomes an $7N$-long vector,

$$
\boldsymbol{X} = \begin{pmatrix} {}^w\bar{q}_1^T & \dots & {}^w\bar{q}_N^T & {}^wp_1^T & \dots & {}^wp_N^T \end{pmatrix}^T
\tag{17}
$$

and the input vector can now be written as

$$
\boldsymbol{U} = \begin{pmatrix} {}^1\bar{\omega}_1^T & \dots & {}^N\bar{\omega}_N^T & {}^1\boldsymbol{v}_1^T & \dots & {}^N\boldsymbol{v}_N^T \end{pmatrix}^T.
\tag{18}
$$

The measurement vector now depends on the two MAVs i and j whose images have been used for computing relative pose information, as well as all altitude measurements:

$$
\boldsymbol{Y}_{i,j} = \left(({}^i\boldsymbol{p}_j{}^v)^T \quad ({}^i\bar{q}_j{}^v)^T \quad {}^wz_1 \quad \dots \quad {}^wz_N \right)^T
\tag{19}
$$

The state and output equations are now

$$
\dot{\boldsymbol{X}} = f(\boldsymbol{X}, \boldsymbol{U} + \boldsymbol{W}) = \begin{pmatrix} \frac{1}{2}\left({}^w\bar{q}_1 \otimes ({}^1\bar{\omega}_1 + \boldsymbol{\eta_{\omega}})\right) \\ \vdots \\ \frac{1}{2}\left({}^w\bar{q}_N \otimes ({}^N\bar{\omega}_N + \boldsymbol{\eta_{\omega}})\right) \\ \boldsymbol{R}({}^w\bar{q}_1)({}^1\boldsymbol{v}_1 + \boldsymbol{\eta_v}) \\ \vdots \\ \boldsymbol{R}({}^w\bar{q}_N)({}^N\boldsymbol{v}_N + \boldsymbol{\eta_v}) \end{pmatrix}
\tag{20}
$$

$$
\boldsymbol{Y}_{i,j} = h_{i,j}(\boldsymbol{X}) + \boldsymbol{V} = \begin{pmatrix} \boldsymbol{R}({}^w\bar{q}_i)({}^wp_i - {}^wp_j)\lambda_{ij} + \boldsymbol{\eta_{p_r}} \\ {}^w\bar{q}_i^* \otimes {}^w\bar{q}_j + \boldsymbol{\eta_{q_r}} \\ {}^wp_{1,z} + \eta_z \\ \vdots \\ {}^wp_{N,z} + \eta_z \end{pmatrix}
\tag{21}
$$

Each update step of the EKF corrects solely the poses of two different MAVs:

$$
\begin{aligned}
\boldsymbol{K}_f &= \boldsymbol{P}_{k+1/k} \boldsymbol{H}_{i,j,k}^T \left(\boldsymbol{H}_{i,j,k} \boldsymbol{P}_{k+1/k} \boldsymbol{H}_{i,j,k}^T + \boldsymbol{R} \right)^{-1} \\
\boldsymbol{X}_{k+1/k+1} &= \boldsymbol{X}_{k+1/k} + \boldsymbol{K}_f\left(\boldsymbol{Y}_{i,j} - h_{i,j}(\boldsymbol{X}_{k+1/k})\right) \\
\boldsymbol{P}_{k+1/k+1} &= (\boldsymbol{I} - \boldsymbol{K}_f \boldsymbol{H}_{i,j,k})\boldsymbol{P}_{k+1/k}
\end{aligned}
\tag{22}
$$

with $\boldsymbol{H}_{i,j,k} = \left.\dfrac{\partial h_{i,j}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{X}_{k/k}}$.

If the pairs of MAVs used in the stereo-vision process are chosen appropriately at each time step (see Section III-A), the poses of all vehicles of the fleet can be updated.

E. Experimental results

The proposed filter has been compared with the monocular AR Drone SLAM described in [2], which is related to the parallel tracking and mapping method (PTAM) [1]. During this experiment with two MAVs, one of the vehicles followed a reference trajectory to draw a one-meter-side square while the other was hovering. The localization computed by our filter, with and without stereo-vision correction, was compared with this monocular method for the moving vehicle (Figure 1, Table I). The results show that the stereo-vision correction significantly improves the localization, with a smaller mean-square error, and the obtained trajectory is consistent with the monocular SLAM (which is however unable to deal with multiple vehicles). Successful experiments have also been conducted for a fleet of three MAVs.
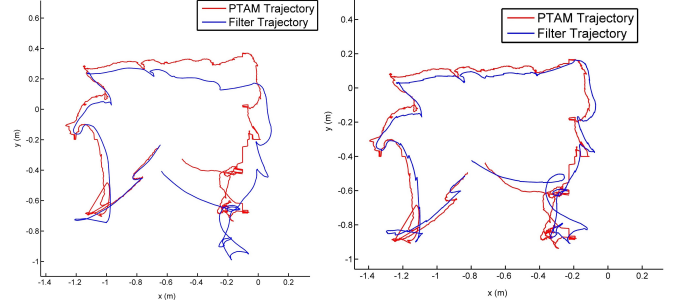

(a) Only-predictive filter   (b) Filter with stereo-vision correction

Fig. 1: Comparison of the filter with monocular SLAM [2]

TABLE I: Filter evaluation (MSE)

Method	x-axis	y-axis
Only-predictive filter	$0.01294\,\mathrm{m}^2$	$0.01522\,\mathrm{m}^2$
Filter with stereo-vision correction	$0.00477\,\mathrm{m}^2$	$0.00456\,\mathrm{m}^2$

An experimental observability analysis has been conducted by studying the rank of the following $(7+N)(7N) \times (7N)$ observability matrix during the estimation process:

$$
\boldsymbol{Obs}_k = \begin{pmatrix} \boldsymbol{H}_k \\ \boldsymbol{H}_k \boldsymbol{F}_{k-7N-1} \\ \boldsymbol{H}_k \boldsymbol{F}_{k-7N-1} \boldsymbol{F}_{k-7N-2} \\ \vdots \\ \boldsymbol{H}_k \boldsymbol{F}_{k-7N-1} \boldsymbol{F}_{k-7N-2} \cdots \boldsymbol{F}_k \end{pmatrix}
\tag{23}
$$

The results showed that the rank of $\boldsymbol{Obs}_k$ is always 2 degrees below its number of columns. Indeed, the lateral translations are not fully observable due to relative pose correction (unlike the vertical axis thanks to the altitude sensor). The filter thus behaves as an odometry on these axes, which requires a good initialization to remain accurate.

III. FORMATION CONTROL

In this section, a formation control is proposed to simultaneously (i) maximize the overlapping fields of view of MAV pairs to guarantee stereo-vision feasibility and (ii) ensure collision avoidance between MAVs. Each MAV has four control inputs: three linear velocities and a yaw angular velocity. To simplify the problem, the control law is divided into three independent parts (see Figure 2):

1) align the MAVs on the same reference yaw,
2) constrain the vehicles in a common reference vertical plane $\mathcal{P}$,
3) maximize the area of overlapping fields of view by pairs of MAVs and avoid collisions.

The first two parts are achieved by simple proportional controllers (for fleet motion, the yaw and vertical plane references can be modified or associated with a MAV leader). The third objective requires a more complex control law, detailed in what follows, to compute the linear velocities $\boldsymbol{u}_i = (u_y \, u_z)^T$ where the y and z axes form an orthogonal basis in the reference plane $\mathcal{P}$.
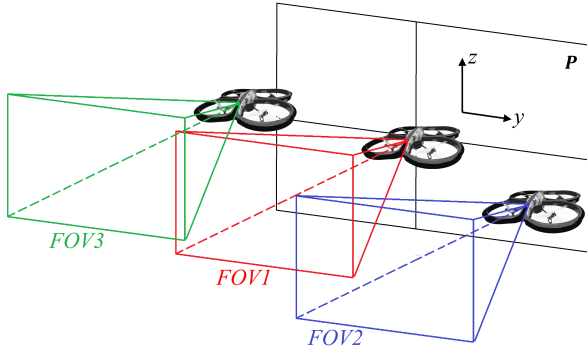

Fig. 2: MAVs are headed in the same direction, in a common vertical plane. The control law aims at maximizing pairs of overlapping fields of view FOV_1, FOV_2 and FOV_3.

A. Maximizing overlapping fields of view

A camera field of view (FOV) can be described as shown in Figure 3, with its width and height parameterized by the depth of field d and the camera angles of view α and β,

$$\begin{cases} w = 2d\tan\left(\dfrac{\alpha}{2}\right) \\ h = 2d\tan\left(\dfrac{\beta}{2}\right) \end{cases}. \quad (24)$$

With this parameterization, the area of overlapping FOVs for MAV pairs can be computed as a function of the MAV positions in the vertical plane, and maximized locally thanks to a gradient ascent algorithm. An algorithm is proposed to

compute this area A_Σ, which is basically the sum of the areas of hatched rectangles in Figure 4.
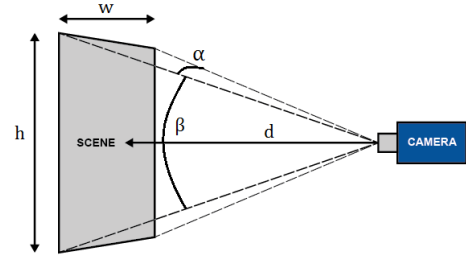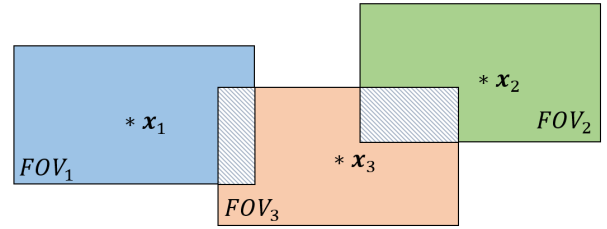

Fig. 3: Field of view of a camera

Fig. 4: Area of overlapping fields of view, to be maximized

The area $A_{i,j}$ of the overlap of two MAV FOVs i and j is equal to

$$\begin{aligned} A_{i,j} = (\min(x_{i,y}, x_{j,y}) - \max(x_{i,y}, x_{j,y}) + w) \\ \cdot (\min(x_{i,z}, x_{j,z}) - \max(x_{i,z}, x_{j,z}) + h) \quad (25) \end{aligned}$$

where $\boldsymbol{x}_i$ is the position of MAV i in $\mathcal{P}$,

$$\boldsymbol{x}_i = \begin{pmatrix} x_{i,y} \\ x_{i,z} \end{pmatrix}. \quad (26)$$

This quantity exists only if the following conditions are true:

$$\begin{cases} \|x_{i,y} - x_{j,y}\| & < & w \\ \|x_{i,z} - x_{j,z}\| & < & h \end{cases}, \quad (27)$$

otherwise $A_{i,j}$ is set to 0. The area function A_Σ can be computed using Algorithm 1, which chooses the minimum number of MAV pairs to have a maximum area of FOV overlaps, while ensuring that all vehicles are taken into account (see Figure 5). This also provides the minimal number of image pairs on which the stereo-vision should be computed, while preserving the connectivity of the fleet.
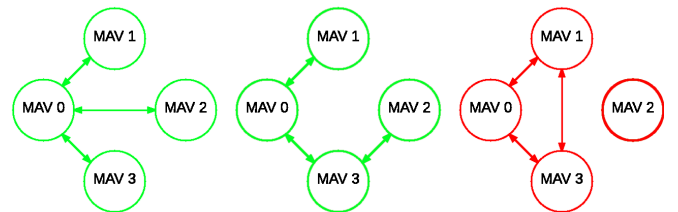

Fig. 5: Algorithm 1 output: last configuration is forbidden

Algorithm 1: Computation of overlapping area function $A_\Sigma(\boldsymbol{x})$

Input : all vehicle positions $\boldsymbol{x}_i, i \in [1, N]$
Output: A_Σ
Data: L initially empty list
Areas for all possible pairs with overlapping FOVs
for $i \leftarrow 1$ **to** $N - 1$ **do**
   for $j \leftarrow i + 1$ **to** N **do**
      if $\|x_{i,y} - x_{j,y}\| < w$ *and* $\|x_{i,z} - x_{j,z}\| < h$ **then**
         add $A_{i,j}$ to L;
      end
   end
end
$A_\Sigma \leftarrow 0$;
A_Σ is the sum of the $N - 1$ largest $A_{i,j}$
such that each MAV is included
for $i \leftarrow 1$ **to** $N - 1$ **do**
   $A_\Sigma \leftarrow A_\Sigma + \max L$;
   remove $\max L$ from L;
end

B. Collision avoidance

The following repulsive term, described in [16], was used to prevent inter-vehicles collisions:

$$\boldsymbol{u}_i^{\text{col}} = 2k \sum_{j=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x}_j)\frac{g_{ij}}{q}, \qquad (28)$$

where k is a positive control gain, $g_{ij} = \exp\left(-\boldsymbol{\delta}_{ij}^T \boldsymbol{\delta}_{ij}/q\right)$, with $\boldsymbol{\delta}_{ij} = (\boldsymbol{x}_i - \boldsymbol{x}_j)$ and q a positive constant determining the repulsion distance.

C. Complete control law

The resulting control input in the vertical plane is obtained by summing the attractive term represented by the maximization of the overlapping FOVs and the repulsive term:

$$\boldsymbol{u}_i = 2k \sum_{j=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x}_j)\frac{g_{ij}}{q} + k'\left(\frac{\partial A_\Sigma(\boldsymbol{x})}{\partial \boldsymbol{x}_i}\right)^T \qquad (29)$$

where
k' is a positive constant,
$\boldsymbol{x} = \left(\boldsymbol{x}_1^T \ \boldsymbol{x}_2^T \ \dots \ \boldsymbol{x}_N^T\right)^T$,
$\frac{\partial A_\Sigma(\boldsymbol{x})}{\partial \boldsymbol{x}_i} = \left(\frac{\partial A_\Sigma(\boldsymbol{x})}{\partial x_{i,y}} \ \frac{\partial A_\Sigma(\boldsymbol{x})}{\partial x_{i,z}}\right)$.
To analyze the stability of this control law, consider the following Lyapunov function:

$$V = \frac{1}{2} \sum_{i=1}^{N} k \sum_{j=1}^{N} g_{ij} + k'(K - A_\Sigma(\boldsymbol{x})), \qquad (30)$$

with K a positive constant such as $K > \max_{\boldsymbol{x}}(A_\Sigma(\boldsymbol{x}))$ to ensure that V is positive. $\dot{V}$ can be expressed as

$$\dot{V} = -k \sum_{i=1}^{N} \sum_{j=1}^{N} \dot{\boldsymbol{\delta}}_{ij}^T \boldsymbol{\delta}_{ij}\frac{g_{ij}}{q} - k'\nabla A_\Sigma(\boldsymbol{x})\dot{\boldsymbol{x}} \qquad (31)$$

with $\nabla A_\Sigma(\boldsymbol{x}) = \left(\frac{\partial A_\Sigma(\boldsymbol{x})}{\partial \boldsymbol{x}_1} \ \frac{\partial A_\Sigma(\boldsymbol{x})}{\partial \boldsymbol{x}_2} \ \cdots \ \frac{\partial A_\Sigma(\boldsymbol{x})}{\partial \boldsymbol{x}_N}\right)$.

Following [17], the double sum can be rewritten as

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \dot{\boldsymbol{\delta}}_{ij}^T \boldsymbol{\delta}_{ij}\frac{g_{ij}}{q} = 2\sum_{i=1}^{N} \sum_{j=1}^{N} (\dot{\boldsymbol{x}}_i)^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\frac{g_{ij}}{q} \qquad (32)$$

Replacing in equation (31) yields

$$\dot{V} = -2k \sum_{i=1}^{N} \sum_{j=1}^{N} (\dot{\boldsymbol{x}}_i)^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\frac{g_{ij}}{q} - k'\nabla A_\Sigma(\boldsymbol{x})\dot{\boldsymbol{x}}$$

$$\dot{V} = -2k \sum_{i=1}^{N} (\dot{\boldsymbol{x}}_i)^T \sum_{j=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x}_j)\frac{g_{ij}}{q}$$
$$-k' \sum_{i=1}^{N} (\dot{\boldsymbol{x}}_i)^T \left(\frac{\partial A_\Sigma(\boldsymbol{x})}{\partial \boldsymbol{x}_i}\right)^T$$

$$\dot{V} = \sum_{i=1}^{N} (\dot{\boldsymbol{x}}_i)^T \left(-2k \sum_{j=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x}_j)\frac{g_{ij}}{q}\right.$$
$$\left. -k'\left(\frac{\partial A_\Sigma(\boldsymbol{x})}{\partial \boldsymbol{x}_i}\right)^T\right) \qquad (33)$$

Since $\dot{\boldsymbol{x}}_i = \boldsymbol{u}_i$, it follows with equation (29) that

$$\dot{V} = \sum_{i=1}^{N} \boldsymbol{u}_i^T \left(-2k \sum_{j=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x}_j)\frac{g_{ij}}{q} - k'\left(\frac{\partial A_\Sigma(\boldsymbol{x})}{\partial \boldsymbol{x}_i}\right)^T\right)$$

$$\dot{V} = \sum_{i=1}^{N} \left(2k \sum_{j=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x}_j)\frac{g_{ij}}{q} + k'\left(\frac{\partial A_\Sigma(\boldsymbol{x})}{\partial \boldsymbol{x}_i}\right)^T\right)^T$$
$$\cdot \left(-2k \sum_{j=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x}_j)\frac{g_{ij}}{q} - k'\left(\frac{\partial A_\Sigma(\boldsymbol{x})}{\partial \boldsymbol{x}_i}\right)^T\right)$$

$$\dot{V} = -\sum_{i=1}^{N} \boldsymbol{u}_i^T \boldsymbol{u}_i \leq 0$$

The derivative $\dot{V}$ of the Lyapunov function V is thus negative semi-definite for the designed control law, which guarantees that the MAVs converge locally to an equilibrium. This corresponds to the situation where $\dot{V} = 0$, i.e. the attraction toward the local maximum of A_Σ is exactly compensated by the repulsion force (28) resulting in a null velocity $\boldsymbol{u}_i = 0$.

D. Results

1) Simulation: Simulations have been performed to analyze the behavior of a larger fleet (see Figure 6). As the maximization is gradient-based, the final layout of the fleet depends on the initial positions of the MAVs.

2) Experiment: An experiment has been made with two MAVs running the entire localization and control loop. Results of this test are presented in Figure 7 and 8. It can be seen that the vehicles converge to a stable formation that maximizes the overlap of their FOVs and respects the collision avoidance distance, which was set to 3 m via parameter q.
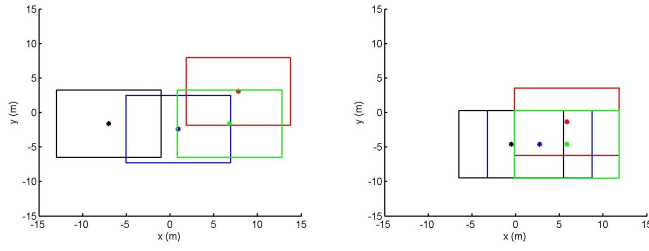
Fig. 6: Simulation with 4 MAVs: positions (dots) and associated FOVs (rectangles) at $t = 0s$ (left) and $t = 10s$ (right).
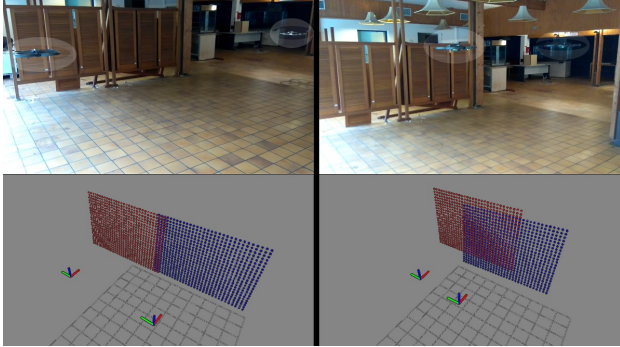

Fig. 7: View of the AR Drones (top) and of the estimated poses (bottom) at $t = 0$s (left) and $t = 8$s (right). Red and blue planes represent the MAV FOVs.
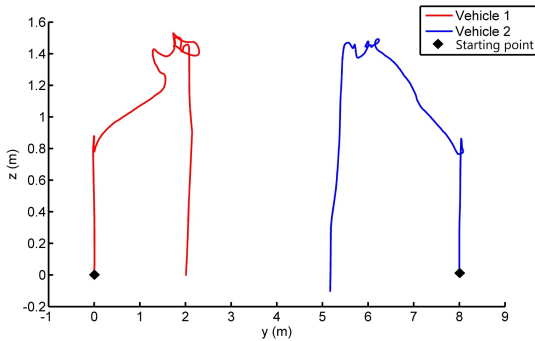

Fig. 8: Experimental MAV trajectories in the y-z plan

IV. CONCLUSIONS AND PERSPECTIVES

In this paper, a sensor fusion scheme including collaborative stereo-vision, IMU, sonar and linear velocity data within an EKF has been proposed to estimate the poses of all the vehicles of a fleet. This algorithm is associated with a control law that maximizes the overlapping fields of view of MAVs in order to improve the stereo-vision process. Experimental results with Parrot® AR Drones show the interest of the complete vision-based estimation and control loop. Stereo-vision was run in real time, which is promising regarding the applicability of this technique for localizing fleets of MAVs in GPS-denied environments.

The localization and control algorithms are designed to incorporate much more than two or three MAVs, which remains to be confirmed by future experimental results. With more MAVs, it could also be interesting to exploit the redundancy of visual sensors to extend the stereo-vision to trifocal (or more) relative pose estimation [18]. Finally, the control law could be extended to perform more elaborate fleet tasks, and the computation load could be distributed if more advanced MAVs are used.

REFERENCES

[1] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan*, 2007, pp. 225–234.

[2] J. Engel, J. Sturm, and D. Cremers, "Scale-aware navigation of a low-cost quadrocopter with a monocular camera," *Robotics and Autonomous Systems*, vol. 62, no. 11, pp. 1646–1656, 2014.

[3] J.-H. Sun, B.-S. Jeon, J.-W. Lim, and M.-T. Lim, "Stereo vision based 3D modeling system for mobile robot," in *Proceedings of the International Conference on Control Automation and Systems, Gyeonggi-do, South Korea*, 2010, pp. 71–75.

[4] M. Sanfourche, V. Vittori, and G. L. Besnerais, "eVO: A realtime embedded stereo odometry for MAV applications," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan*, 2013, pp. 2107–2114.

[5] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 571–583, 2007.

[6] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular SLAM with multiple micro aerial vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan*, 2013, pp. 3962–3970.

[7] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 354–366, 2013.

[8] G. Bresson, R. Aufrere, and R. Chapuis, "Consistent multi-robot decentralized SLAM with unknown initial positions," in *Proceedings of the 16th International Conference on Information Fusion, Istanbul, Turkey*, 2013, pp. 372–379.

[9] M. W. Achtelik, S. Weiss, M. Chli, F. Dellaert, and R. Siegwart, "Collaborative stereo," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA*, 2011, pp. 2242–2248.

[10] E. Montijano, D. Zhou, E. Cristofalo, M. Schwager, and C. Sagues, "Vision-based distributed formation control without a global reference frame," *International Journal of Robotics Research*, 2014.

[11] P.-J. Bristeau, F. Callou, D. Vissiere, and N. Petit, "The navigation and control technology inside the AR drone micro UAV," in *Proceedings of the 18th IFAC world congress, Milan, Italy*, vol. 18, no. 1, 2011, pp. 1477–1484.

[12] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[13] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE International Conference on Computer Vision, Kerkyra, Greece*, vol. 2, 1999, pp. 1150–1157.

[14] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[15] L. Kneip and P. Furgale, "OpenGV: A unified and generalized approach to real-time calibrated geometric vision," in *Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China*, 2014, pp. 1–8.

[16] A. Kahn, J. Marzat, H. Piet-Lahanier, and M. Kieffer, "Cooperative estimation and fleet reconfiguration for multi-agent systems," *Proceedings of the IFAC Workshop on Multivehicle Systems, Genova, Italy*, pp. 11–16, 2015.

[17] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, no. 10, pp. 2406–2411, 2009.

[18] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.