Self-positioning of a team of flying smart cameras

Fabio Poiesi and Andrea Cavallaro
Centre for Intelligent Sensing, Queen Mary University of London
Email: {fabio.poiesi,a.cavallaro}@qmul.ac.uk

***Abstract*—Quadcopters are highly maneuverable and can provide an effective means for an agile dynamic positioning of sensors such as cameras. In this paper we propose a method for the self-positioning of a team of camera-equipped quadcopters (flying cameras) around a moving target. The self-positioning task is driven by the maximization of the monitored surface of the moving target based on a dynamic flight model combined with a collision avoidance algorithm. Each flying camera only knows the relative distance of neighboring flying cameras and its desired position with respect to the target. Given a team of up to 12 flying cameras, we show they can achieve a stable time-varying formation around a moving target without collisions.**

I. Introduction

Advances in robotics and embedded systems bring together a powerful means for the development of moving smart-cameras [1], [2]. In particular, smart cameras can be dynamically positioned in an environment by mounting them on quadrotors to best achieve an assigned task, such as best camera coverage [3], [4], cooperative surveillance [5], [6], target tracking [7], [8] or relative localization in swarms [9], [10]. We will refer to camera-equipped quadcopters as *flying cameras*.

The positioning of flying cameras can be carried out using external localization devices such as GPS or motion capture systems (e.g. VICON) [11], [12] or independently (autonomous sensing) via sensing and on-board processing [10]. The independent positioning (self-positioning) of flying cameras using only on-board processing needs to account for sensing inaccuracies that affect the control mechanisms and the physical dynamics of the robotic platform that may in turn affect the sensing. Self-positioning requires *relative localization*, which can be achieved by using geometric patterns stuck on the flying cameras. These patterns can either be circular [9] or square [10], and can be processed by cameras at 60Hz with a precision of a few centimeters for distances of meters.

Flying cameras are generally simulated assuming oscillation-free quadrotors [5], [6]. However, oscillations are frequent and caused by the nature of the quadrotors' dynamics. To reduce oscillations in real-world applications, cameras can be mounted on gimbals [13], which might not be sufficient to stabilize a camera in the case of abrupt and vigorous maneuvers. Therefore, quadrotor's oscillations need to be taken into account when using on-board directional sensing, such as for target tracking. The task of *target tracking* with multiple flying cameras aims at maintaining the observability of a moving target over time [7]. Each flying camera autonomously detects the target and communicates the measurements to the other flying cameras. The target position is shared with respect to a global coordinate system and its direction with respect to each other. A Kalman filter can be used to estimate the target state. Target tracking from flying cameras using on-board processing can exploit the target orientation and position to self position the cameras around a target [14]. Other approaches aim at maximizing either the number of targets to track, subject to a desired quality, or the duration of the high-quality tracking for all targets [8].

In this paper we focus on the problem of the simultaneous positioning of multiple flying cameras while accounting for real-world dynamics. We combine the dynamic model of quadrotors [15], [16] with their coordinated self-positioning [14], and extend it for the problem of directional camera views. To allow each flying camera to reach a desired (goal) position from a random spatial initialization, each camera computes its own direction of motion relying on the position and orientation of the detected moving target, and on the dynamics of the neighboring cameras in order to avoid collisions. To the best of our knowledge, ours is the first work to address the sensing problem for teams of smart cameras mounted on quadrotors with real-world dynamics. Indeed, flying cameras are usually assumed to hover with stable dynamics and oscillations are not considered [6]. Moreover, our approach differs from [14] because the solution obtained with the Kabsch algorithm [17] depends on the initialization.

The paper is organized as follows. In Sec. II we formulate the problem of tracking a target with a team of flying cameras in formation. Section III describes the proposed approach for the self-positioning of flying cameras. In Sec. IV we evaluate our approach using simulations and in Sec.V we draw the conclusions.

II. Problem formulation

Let us consider N flying cameras (i.e. smart cameras mounted on quadrotors) with angle of view ϕ, known intrinsic parameters and thus capable of retrieving the distance of an object of known size, the target, up to a distance h. We assume that flying cameras mount a 3D proximity sensor capable of estimating the distance of neighboring flying cameras [18]. We aim at positioning the N cameras in order to achieve the desired view of a target. The desired view is defined in this paper as the full inclusion of the target in the field of view of each flying camera.

Let $M(k) = \{q_i(k)\}_{i=1}^{N}$ represent the team of flying cameras at time k, where $q_i(k)$ is the state of the i^{th} camera at k, and $k = 0, ..., K$. The state $q_i(k) = (x_i(k), R_i(k))$ defines the position $x_i(k) \in \mathbb{R}^3$ in the global coordinate system and attitude $R_i(k) \in SO(3)$ [15]. $R_i(k)$ is the rotation matrix from the local (body) to the global coordinate system that defines

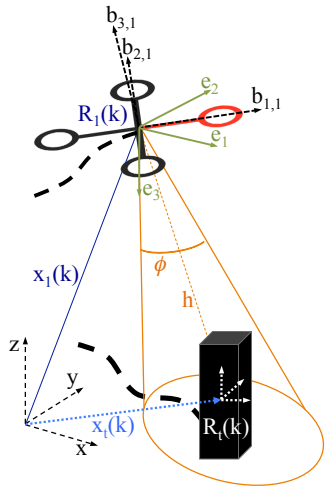

Fig. 1. Coordinate systems of a flying camera and a target with respect to the global coordinates. The cone models the field of view of the camera with view angle ϕ and capture distance h.

the flying camera's attitude. The body is defined by three main directions $b_{1,i}(k) = R_i(k)e_1$, $b_{2,i}(k) = R_i(k)e_2$ and $b_{3,i}(k) = -R_i(k)e_3$ where $e_1 = (1,0,0)^T$, $e_2 = (0,1,0)^T$ and $e_3 = (0,0,1)^T$. The flying camera's motion is determined by a controller that follows a desired trajectory $X_{d,i} = \{x_{d,i}(k)\}_{k=0}^{K}$, with $x_{d,i}(k) \in \mathbb{R}^3$, and a desired direction of the first body direction $b_{1_d,i}(k)$ [15]. Fig. 1 summarizes the variables.

The target trajectory is defined as $X_t = \{x_t(k)\}_{k=0}^{K}$, where $x_t(k) \in \mathbb{R}^3$ is the position of the target in the global coordinate system. The desired (goal) locations of each flying camera are defined by the set $G(k) = \{g_i(k)\}_{i=1}^{N}$, where $g_i(k) \in \mathbb{R}^3$. The relative goal locations are known to the flying cameras [14]. The target state is $t(k) = (x_t(k), R_t(k), Q_t)$, where $R_t(k)$ defines the target attitude and Q_t defines the 3D bounding volume occupied by the target.

Let us assume that the target location $x_t(k)$ is known by each flying camera with respect to its local coordinate system. The target attitude $R_t(k)$ can be computed using visual features (e.g. image gradients) and motion features (e.g. temporal changes of SIFT features) to estimate orientation changes over time [19]. Each camera self-positions on its goal location $g_i(k)$ to cover a desired view of the target while avoiding collisions with neighbors [16].

III. Self-positioning and collision avoidance

A. Force-based model

The behavior of the flying cameras is modeled using two main forces, a force towards the goal location, $F_{g,i}(k) \in \mathbb{R}^3$, and a repulsive force, $F_{r,i}(k) \in \mathbb{R}^3$, from neighboring flying cameras. The total force $F_{m,i}(k)$ acting on the i^{th} camera is defined as

$$F_{m,i}(k) = F_{g,i}(k) + F_{r,i}(k). \quad (1)$$

$F_{m,i}(k)$ affects the desired trajectory $x_{d,i}(k)$ that $q_i(k)$ follows and makes the flying camera changing its attitude by recomputing a new desired trajectory and a new body direction.

The attraction to the goal location is independently computed by each camera, for example based on the individual observation of the target and/or the environment [7]. The attractive force to the goal is defined as

$$F_{g,i}(k) = K_g \frac{L_{i,g}(k)}{||L_{i,g}(k)||} + D_g \frac{dL_{i,g}(k)}{dk}, \quad (2)$$

where $L_{i,g}(k) \in \mathbb{R}^3$ is the relative vector from $q_i(k)$ and the goal location $g_i(k)$. K_g and D_g are constants [16].

As mentioned in the previous section, flying cameras are assumed to be capable of estimating the distance of neighbors within a certain range ρ [18]. The rules for avoiding collisions are implemented independently in each camera using repulsion forces. Repulsion forces determine the behavior of a camera and control its attitude when other cameras are located within the range ρ. The repulsive force $F_{r,i}(k)$ acting on $q_i(k)$ is

$$F_{r,i}(k) = \sum_{j=1, j\neq i}^{N} w_{i,j}(k) F_{r,i,j}(k), \quad (3)$$

where $F_{r,i,j}(k) \in \mathbb{R}^3$ is the repulsive force between $q_i(k)$ and $q_j(k)$, and $w_{i,j}(k)$ is a distance weight function defined as

$$w_{i,j}(k) = \frac{1}{\exp\left(aL_{i,j}(k) - b\right) + c} + \frac{1}{\exp\left(0.5aL_{i,j}(k) - b\right) + c}, \quad (4)$$

where a, b, c are constants [16] and $L_{i,j}(k) \in \mathbb{R}^3$ is the vector between $q_i(k)$ and $q_j(k)$. $L_{i,j}(k)$ can be calculated by the on-board vision systems [10]. $F_{r,i,j}(k)$ is defined as

$$F_{r,i,j}(k) = K_d(||L_{i,j}(k)|| - L_r)L_{i,j}(k) + D_d \frac{dL_{i,j}(k)}{dk}, \quad (5)$$

where L_r is the desired inter-distance among flying cameras that enables stabilization of the formation, and K_d and D_d are constants [16].

B. Target-based positioning

With respect to its local coordinates system, $q_i(k)$ aims at reaching the position

$$\begin{aligned} g_{l,i}(k) &= R_t(k)\, g_i(k) \\ &= R_t(k)\, (x_t(k) + x_{f,i}), \end{aligned} \quad (6)$$

where $x_{f,i}$ is the position with respect to $x_t(k)$. Note that the position to reach depends on the target's orientation $R_t(k)$.

Each flying camera computes its own velocity that minimizes the required time to reach the goal location. The camera's velocity is locally calculated as

$$v_{l,i}(k) = \beta(k)\, (R_i(k)\, R_t^{-1}(k)\, g_{l,i}(k) - R_i(k)\, x_i(k)), \quad (7)$$

which is independent of the global coordinate system. $\beta(k)$ is a time-dependent weighting function defined as

$$\beta(k) = \begin{cases} 1 - \frac{\tau - k}{\tau - k_0} & \text{if } k \in [0, \tau] \\ 1 & \text{if } k \in (\tau, K], \end{cases} \quad (8)$$

where τ is the deadline to reach a goal location and k_0 is the initialization time.

At each time step, a flying camera computes its next location in order to determine the attitude and thrust of the

quadrotor. When there are multiple flying cameras, the desired trajectories need to be recomputed by taking into consideration neighboring cameras. The desired trajectory of a camera is computed as

$$x_{d,i}(k) = x_i(k-1) + \left(R_i(k)\ v_{l,i}(k) + \frac{F_{m,i}(k)}{m_i}\, dk \right) dk, \quad (9)$$

where $F_{m,i}(k)$ is the total force acting on the i^{th} flying camera that depends on the position of the neighbors and $m_i \in \mathbb{R}$ is the camera's weight.

Since the desired heading direction points towards the next desired location, the desired first-body direction is

$$b_{1_d,i}(k) = x_{d,i}(k) - x_i(k-1). \quad (10)$$

C. Flight dynamics model

The dynamics of the camera-equipped quadrotors are determined by four identical propellers, which are equidistant from the centre of body, and generate a thrust and torque orthogonal to the plane defined by $b_{1,i}(k)$ and $b_{2,i}(k)$ [15]. The motion of the i^{th} camera can be expressed as[1]

$$\begin{aligned} \dot{x}_i &= v_i \\ m_i \dot{v}_i &= m_i g e_3 - f_i R_i e_3 \\ \dot{R}_i &= R_i \hat{\Omega}_i \\ J_i \dot{\Omega}_i + \Omega_i \times J_i \Omega_i &= M_i, \end{aligned} \quad (11)$$

where $v_i \in \mathbb{R}^3$ is the velocity in the global coordinate system, $g = 9.8m/s^2$ is the gravity acceleration, $f_i = \sum_{n=1}^{4} f_{n,i} \in \mathbb{R}^3$ is the total thrust generated by the four propellers in $b_{3,i}(k)$ direction (Fig. 1), $\Omega_i \in \mathbb{R}^3$ is the angular velocity, $J_i \in \mathbb{R}^{3\times3}$ is the inertia matrix, $M_i \in \mathbb{R}^3$ is the total moment (the last three terms are calculated in the local coordinate system). The *hat operator* is defined as $\hat{x}y = x \times y$ for all $x, y \in \mathbb{R}^3$, where $\times$ is the cross product [15].

Given a transitional control command, the camera's dynamics are controlled by the magnitude of the total thrust f_i and by the control moment $M_i = (M_{1,i}, M_{2,i}, M_{3,i})^T$ for the desired attitude:

$$\begin{bmatrix} f_i \\ M_{1,i} \\ M_{2,i} \\ M_{3,i} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -d & 0 & d \\ d & 0 & -d & 0 \\ -c_\tau & c_\tau & -c_\tau & c_\tau \end{bmatrix} \begin{bmatrix} f_{1,i} \\ f_{2,i} \\ f_{3,i} \\ f_{4,i} \end{bmatrix}, \quad (12)$$

where d is the distance from the center of the flying camera to the center of each rotor and c_τ is a constant.

Given a trajectory point $x_{d,i}(k)$, the total thrust f_i and the desired direction of the third-body axis $b_{3_d,i}(k)$ are selected to stabilize the translational dynamics. $b_{2_d,i}(k)$ is then computed as a function of $b_{1_d,i}(k)$ and $b_{3_d,i}(k)$. Please refer to [15] for a detailed explanation of the control model.

As an example, in Fig. 2 a team of flying cameras self-positions around the moving target (black cuboid) by reaching their own goal locations (numbered black circles). The field of view of each camera is shown with cones. In Fig. 2a the flying cameras are initialized in random locations and their desired formation is an octagon. The flying cameras' dynamics make

[1] k is temporally omitted for simplicity of notation.

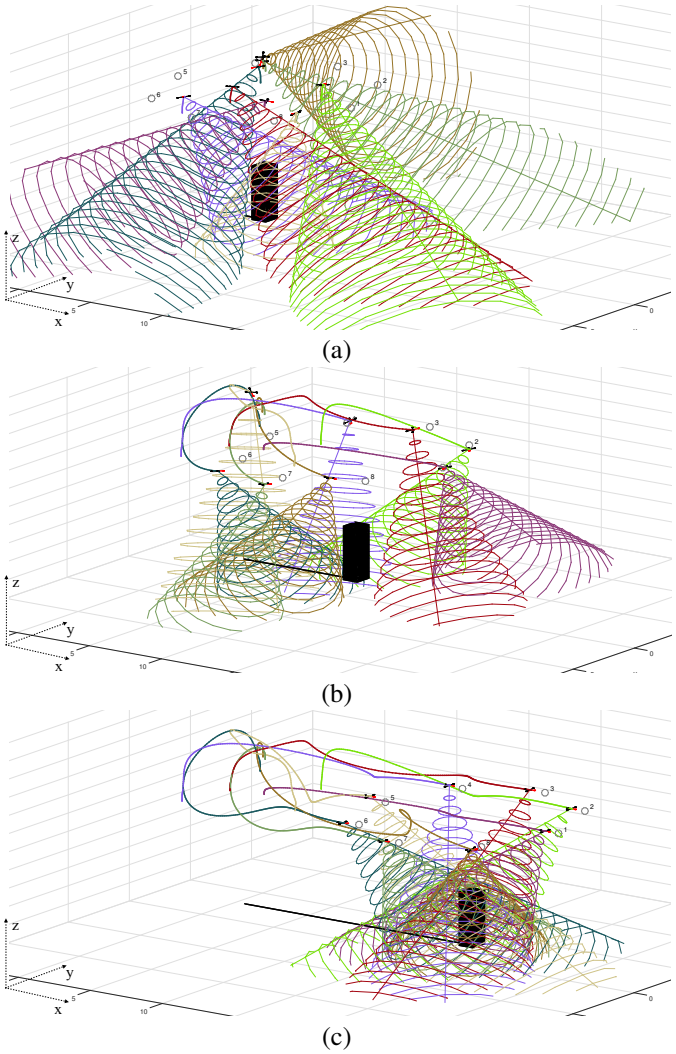

Fig. 2. A simulation result showing how the flying cameras stabilize in the direction of the target's motion. (a) Cameras are initialized and head towards the goal locations; (b) cameras are about to reach the formation; (c) cameras reach their goal locations with a stable formation.

the observation of a target difficult during the stabilization of the formation because of the limited field of view. When the cameras are about to reach the goal locations (Fig. 2b), the formation becomes more stable and the target is almost fully covered by the flying cameras' field of views. In Fig. 2b,c the cameras correctly reach their own goal locations by following and fully viewing the target.

IV. Simulation Results

In this section we show how real-world flight dynamics of teams of flying cameras affect the viewing of a target during tracking. The starting locations of the cameras are randomly initialized for each simulation. We assess the stability and view performance using $N = 6, 8, 10, 12$. The parameters of the camera's dynamics are set as in [16], except those in Eq. 4 that we set as $a = 4$, $b = 1$, $c = 0.6$ in order to allow cameras to have a smoother repulsion when they interact. The localization range is $\rho = 8m$. The goal locations $G(k)$ are relative to the target location and are known a priori by the cameras (Table I).

TABLE I. DESIRED (GOAL) LOCATIONS WITH RESPECT TO THE MOVING TARGET LOCATION. GOAL LOCATIONS ARE GIVEN AS PRIOR TO FLYING CAMERAS IN OUR EXPERIMENTS. KEY - FC: FLYING CAMERAS; N: NUMBER OF FLYING CAMERAS; L: RADIUS OF THE FORMATION; H: FLIGHT ALTITUDE.

$x_{f,i} = (L \cdot cos((i-1)\frac{\pi}{N/2}, L \cdot sin((i-1)\frac{\pi}{N/2}, H)^T$

Name	Parameters
FC-6	$N = 6, L = 6, H = 7$
FC-8	$N = 8, L = 6, H = 7$
FC-10	$N = 10, L = 6, H = 7$
FC-12	$N = 12, L = 6, H = 7$

The moving target is approximated as a cuboid. The deadline to reach the formation is $\tau = 3s$ and $dk = 0.04s$. Each camera is pointed towards the target location with respect to the goal location of the flying camera, and the parameters are $\phi = \pi/4$ and $h = 20m$.

We analyze the positional error of the cameras by calculating the Euclidean distance over time between each camera and its desired location, such as $d_{xg,i}(k) = ||x_i(k) - g_i(k)||_2$. The formation is considered reached when the distance between a flying camera and its goal location remains less than $\epsilon = 0.001m$ till the end of the experiment. We quantify the camera view performance by counting the number of vertices (from 1 to 8) of the moving target that are included in the field of view over time by each flying camera (i.e. considering the target as a wireframe). Formation stability and view performance are quantified by running 100 times each experiment with different random initializations of the cameras' position, and by calculating the resulting mean and standard deviation values. We assess the view performance under two target dynamics: a straight line and a sinusoidal trajectory for the target.

Fig. 3 shows a target moving on a straight line when the number of flying cameras forming a team increases. The velocity of the target is $v_{tar} = (2 \cdot dk, 0, 0)^T$ (straight line along the x direction). The time required to reach the formation stability increases by increasing the number of flying cameras. There are also more oscillations when the number of cameras increases because of the adjustment of the heading direction of the flying cameras with respect to their own time-varying goal locations.

Fig. 4 quantifies the stabilization time in terms of mean and standard deviation on a set of repeated simulations. Given the same radius of the formation, the larger the number of cameras, the longer the average time required for each camera to fully view the target. On average, the stabilization time has similar values throughout the experiments, except for 12 flying cameras for which their standard deviation is larger than the other cases. These oscillations also affect the viewing of the target (Fig. 5). Given the same space/distance, the view performance has a common pattern among all the experiments. In particular, the V-like shapes shown in Fig. 5a,c,e,g are due to the flying camera arrangement within the formation. Flying cameras that have to reach goal locations in front of the target need to cover on average a longer path to traverse and are therefore more likely to interact with other cameras (Fig. 5b,d,f,h).

Fig. 6 shows the results obtained with different numbers of cameras viewing the target with a sinusoidal trajectory. The velocity of the target is $v_{tar} = (2 \cdot dk, 0.08 \cdot cos(\frac{2}{5}\pi k), 0)^T$.

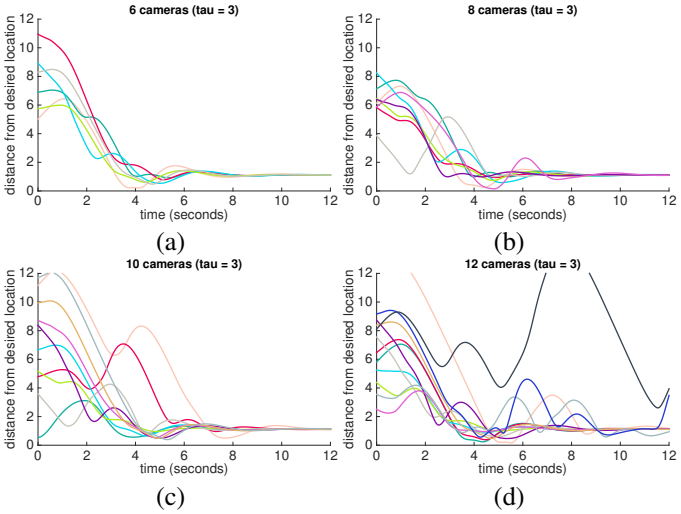

Fig. 3. Distance between flying cameras and desired (goal) locations over time when increasing the number of cameras involved in the formation: (a) $N = 6$, (b) $N = 8$, (c) $N = 10$ and (d) $N = 12$. Given the same formation radius, the time to reach a stable formation increases with the number of cameras. In particular, with $N = 12$ the higher number of cameras leads to instabilities due the increased probability of interactions among cameras.

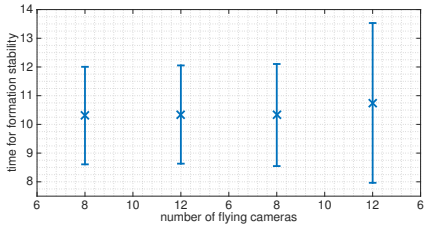

Fig. 4. Time required to reach stability with four different team formations. The values (obtained with 100 runs of randomly initialized simulations) show mean (cross symbols) and standard deviation (vertical segments).

Similarly to the previous case, some cameras achieve better view performance than others. In particular, the cameras located on the sides of the target have on average the best view performance (Fig. 6a,c,e,g). Conversely, the front and the rear cameras view the target for shorter intervals. The cameras require on average 8 seconds to converge to the formation, then the view pattern becomes periodic. When the target changes direction, the cameras have to adapt their dynamics in order to keep viewing it and this introduces a delay. By increasing the number of cameras, when the target changes direction, interactions between cameras are more likely and therefore the time needed to achieve a certain formation increases. Fig. 7 shows examples where the tracking of a target with sinusoidal trajectory leads to large flight dynamics changes, which in turn cause unstable views of the target.

V. CONCLUSIONS

We presented an approach for the self-positioning of teams of flying smart cameras that accounts for real-world flight dynamics. The proposed approach employs the flight model of a quadrotor [15] using a tracking controller to follow a desired trajectory along with a desired heading direction. The flying camera control also models a collision avoidance algorithm [16] that senses the position of neighbors. The

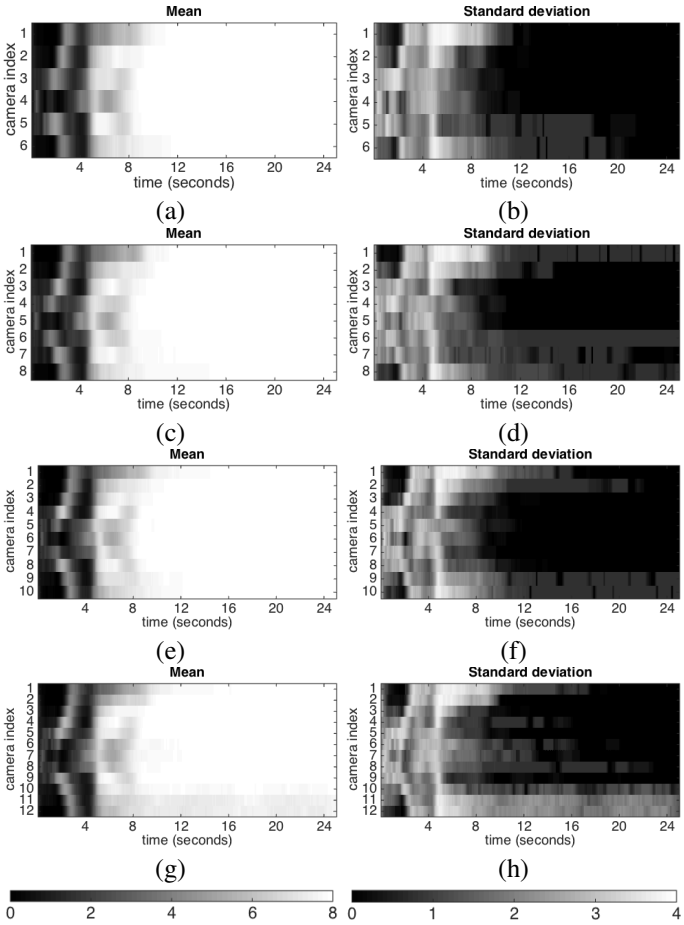

Fig. 5. Camera view performance using formations with an increasing number of flying cameras and a cuboid-shaped target moving with a straight-line trajectory. (a,b) $N = 6$; (c,d) $N = 8$; (e,f) $N = 10$; (g,h) $N = 12$. The target is defined by 8 vertices and the results show the number of vertices viewed over time. The (a,c,e,g) mean and (b,d,f,h) standard deviation are calculated on a set of 100 experiments for each case. (a,c,e,g) The gray scale indicates the mean number of vertices viewed by each camera (i.e. considering the target as a wireframe). (b,d,f,h) The gray scale indicates the standard deviation of the number of vertices viewed by the flying cameras.

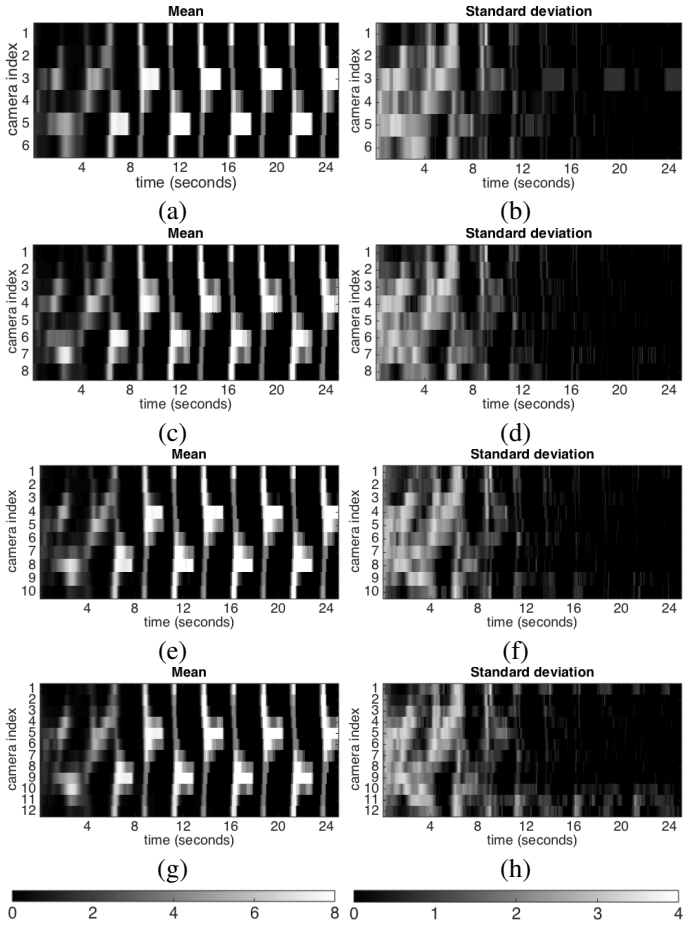

Fig. 6. Camera view performance using formations with an increasing number of flying cameras and a cuboid-shaped target moving with a sinusoidal trajectory. (a,b) $N = 6$, (c,d) $N = 8$, (e,f) $N = 10$, (g,h) $N = 12$. The target is defined by 8 vertices and the results show the number of vertices viewed over time. The (a,c,e,g) mean and (b,d,f,h) standard deviation are calculated on a set of 100 experiments for each case. (a,c,e,g) The gray scale indicates the mean number of vertices viewed by each camera (i.e. considering the target as a wireframe). (b,d,f,h) The gray scale indicates the standard deviation of the number of vertices viewed by the flying cameras.

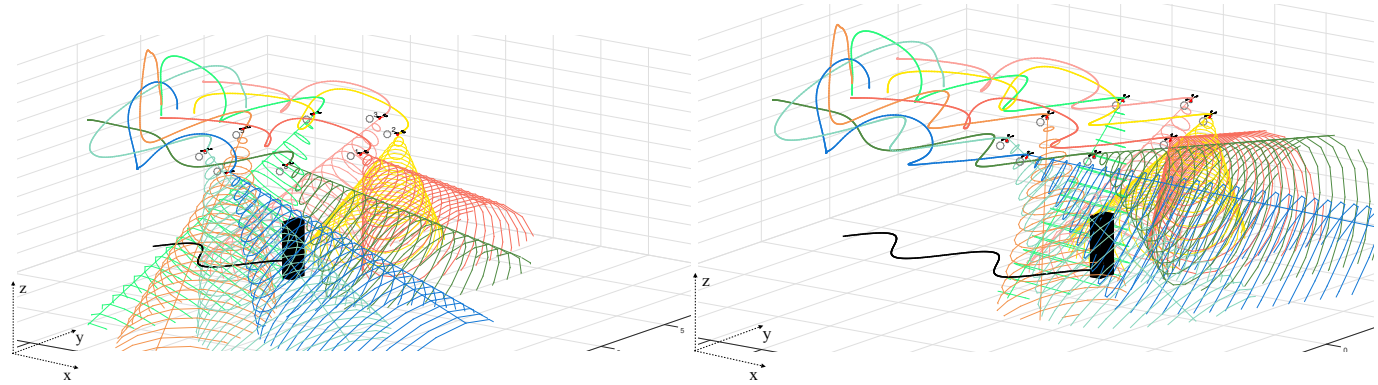

Fig. 7. Flying cameras tracking a target moving with a sinusoidal trajectory. (a) The maneuver of the flying cameras while following the target. (b) Unstable condition reached by the cameras (that fail to fully view the target).

desired formation is reached by relying only on observations made directly by the camera about the attitude of a moving target (e.g. position and orientation). Simulations showed that flying cameras become unstable in the case of tight formations and targets moving with varying directions. Moreover, cameras that in the formation need to reach a position in front of the target are more subject to oscillations than the others, because they have a longer path to traverse and a limited/constrained space for their movement.

Future research will involve the extension of the control algorithm to model advanced formations that predict target dynamics to have the best view during target tracking.

ACKNOWLEDGMENTS

This work was supported by the Artemis JU and the UK Technology Strategy Board through COPCAMS Project under Grant 332913.

REFERENCES

[1] G. Chmaj and H. Selvaraj, "Distributed processing applications for UAV/drones: a survey," in *Proc. of International Conference on Systems Engineering*, Las Vegas, NV, USA, Aug. 2014, pp. 449–454.

[2] J.C. SanMiguel, C. Micheloni, K. Shoop, G.L. Foresti, and A. Cavallaro, "Self-reconfigurable smart camera networks," *IEEE Computer*, vol. 47, no. 5, pp. 67–73, May 2014.

[3] M. Schwager, B.J. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of IEEE*, vol. 99, no. 9, pp. 1541–1561, Sep. 2011.

[4] L. Doitsidis, S. Weiss, A. Renzaglia, M.W. Achtelik, E. Kosmatopoulos, R. Siegwart, and D. Scaramuzza, "Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 173–188, Aug. 2012.

[5] M. Saska, J. Chudoba, L. Preucil, J. Thomas, G. Loianno, A. Tresnak, V. Von, and V. Kumar, "Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance," in *Proc. of International Conference on Unmanned Aircraft Systems*, Orlando, FL, USA, May 2014.

[6] J. Hu, L. Xie, J. Xu, and Z. Xu, "Multi-agent cooperative target search," *Sensors*, vol. 14, no. 6, pp. 9408–9428, Jun. 2014.

[7] B. Bethke, M. Valenti, and J. How, "Cooperative vision based estimation and tracking using multiple UAVs," in *Proc. of International Conference on Cooperative Control and Optimization*, Gainesville, FL, USA, Jan. 2007, pp. 179–189.

[8] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *Proc. of Internation Conference on Intelligent Robots and Systems*, Chicago, IL, USA, Sep. 2014, pp. 3067–3072.

[9] J. Faigl, T. Krajnik, J. Chudoba, L. Preucil, and M. Saska, "Low-cost embedded system for relative localization in robotic swarms," in *Proc. of International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 993–998.

[10] T. Nageli, C. Conte, A. Domahidi, M. Morari, and O. Hilliges, "Environment-independent formation flight for micro aerial vehicles," in *Proc. of International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, Sep. 2014.

[11] G. Vasarhelyi, C. Viragh, G. Somorjai, N. Tarcai, T. Szorenyi, T. Nepusz, and T. Vicsek, "Outdoor flocking and formation flight with autonomous aerial robots," in *Proc. of International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, Sep. 2014.

[12] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, Apr. 2012.

[13] L. Sun, S. Baek, and D. Pack, "Distributed probabilistic search and tracking of agile mobile ground targets using a network of unmanned aerial vehicles," in *Human behavior understanding in networked sensing*, P. Spagnolo, P.L. Mazzeo, and C. Distante, Eds., pp. 301–319. Springer, 2014.

[14] M. Aranda, G. Lopez-Nicolas, C. Sagues, and M.M. Zavlanos, "Three-dimensional multirobot formation control for target enclosing," in *Proc. of International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, Sep. 2014.

[15] T. Lee, M. Leok, and N.H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *Proc. of International Conference on Robotics and Automation*, Atlanta, GA, USA, Dec. 2010, pp. 5420–5425.

[16] M. Saska, J. Vakula, and L. Preucil, "Swarms of micro aerial vehicles stabilized under a visual relative localization," in *Proc. of International Conference on Robotics and Automation*, Hong Kong, China, May 2014, pp. 3570–3575.

[17] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A*, vol. 32, no. 5, pp. 922–923, Sep. 1976.

[18] J.F. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano, "3-D relative positioning sensor for indoor flying robots," *Autonomous Robots*, vol. 33, no. 1-2, pp. 5–20, Aug. 2012.

[19] O. Ozturk, T. Yamasaki, and K. Aizawa, "Estimating human body and head orientation change to detect visual attention direction," in *Proc. of Asian Conference on Computer Vision Workshop*, Queenstown, NZ, Nov. 2010, pp. 410–419.