Citation: M. Aranda, G. López-Nicolás, C. Sagüés, Y. Mezouar. Formation control of mobile robots using multiple aerial cameras. IEEE Transactions on Robotics, vol. 31, no. 4, pp. 1064-1071, 2015

Formation Control of Mobile Robots using Multiple Aerial Cameras

Miguel Aranda, *Member, IEEE*, Gonzalo López-Nicolás, *Senior Member, IEEE*, Carlos Sagüés, *Senior Member, IEEE*, and Youcef Mezouar

***Abstract*—This paper describes a new vision-based control method to drive a set of robots moving on the ground plane to a desired formation. As the main contribution, we propose to use multiple camera-equipped Unmanned Aerial Vehicles (UAVs) as control units. Each camera views, and is used to control, a subset of the ground team. Thus, the method is partially distributed, combining the simplicity of centralized schemes with the scalability and robustness of distributed strategies. Relying on a homography computed for each UAV-mounted camera, our approach is purely image-based and has low computational cost. In the control strategy we propose, if a robot is seen by multiple cameras, it computes its motion by combining the commands it receives. Then, if the intersections between the sets of robots viewed by the different cameras satisfy certain conditions, we formally guarantee the stabilization of the formation, considering unicycle robots. We also propose a distributed algorithm to control the camera motions that preserves these required overlaps, using communications. The effectiveness of the presented control scheme is illustrated via simulations and experiments with real robots.**

I. Introduction

Multiagent systems are very interesting in robotics due to their ability to perform complex tasks with great efficiency and reliability. In this context, we address in this paper the problem of bringing a set of ground mobile robots to a desired geometric configuration, which is also referred to as formation stabilization. Typically, the formation to be stabilized is defined in terms of absolute positions for the robots to reach [1], [2], or as relative position vectors or distances between the agents [3]. Between the two latter varieties, distance-based formation control [4], [5] employs simpler information and does not require a global reference frame for the robots, while relative position-based methods [6], [7] exhibit stronger stability properties. In this and other related multirobot problems [8], [9], distributed control strategies tend to be preferred, for robustness and scalability.

A very relevant characteristic of the method we present is the use of vision. Cameras are powerful and affordable sensors that have been, and continue to be, extensively used for control tasks [10]. In the field of multirobot systems, [11] is an early example of a vision-based framework to control a formation, while [12], [13] tackle distributed motion coordination tasks. [14] considers visibility constraints and environmental obstacles in leader-follower formations. In these works, the cameras are carried by the robots, whereas the work [15], whose core idea is more closely related to our method, employs a camera-equipped aerial robot as a supervisory unit. This unit is used to compute the absolute localization of an ensemble of ground robots and to control the members to form a grossly-modeled shape bounded by an ellipse. Multiple-view geometric models have proven valuable to increase the robustness of performance in control schemes, and have been applied to multirobot control scenarios. In this respect, the work [16], which is at the origin of the one we present, proposes a homography-based centralized system with a single aerial camera. Here, multiple aerial cameras are employed. Using a current image of a subset of the ground robots and a reference one, each camera computes a transformation that creates a set of *desired* image points, from which it defines desired motion objectives, which are transmitted to the robots. Then, each robot computes its actual control input using this information, received from one or multiple sources. Thus, our system is not centralized (since the control commands are generated from *partial* information) and does not use a global reference frame. Formation control is challenging under these conditions [4], [5]. We propose a strategy for the integration of commands and we define constraints for the intersections between the sets of robots viewed by the cameras that guarantee the stabilization of the global formation.

In our method, the visual information is obtained by aerial cameras carried by UAVs acting as control units, whose motion needs to be controlled to ensure that the visibility of the ground robots is maintained. This has elements in common with works on visual coverage, such as [17], which presents a distributed algorithm for positioning multiple cameras over a planar area to be covered. The work [18] studies the problem of providing protection to a ground convoy with a set of UAVs, considering the ground vehicles are stationary or move along straight lines. In our case, the motion of the UAVs must preserve the coverage of the robots (i.e. of a dynamic environment) and is further constrained due to the need to maintain robot intersections. It is challenging to address this problem from the standpoint of optimality. We focus here on effectiveness, and propose a distributed algorithm that ensures the requirements are met.

The use of external cameras for the control task we address has the advantage of allowing the robots to carry simpler onboard equipment, as they do not need to handle costly sensing and computation. In addition, they do not have to transmit information, which typically consumes a lot of power. In particular, the main contribution of this paper is the use of multiple cameras, which improves the maximum coverage, robustness and scalability with respect to a single-camera setup. Our method is partially distributed, preserving some properties of centralized systems (e.g. more efficient performance than distributed controllers). The system can be flexibly dimensioned by selecting the appropriate number of cameras for a given number of robots and size of the workspace. Our approach is image-based and, therefore, does not use range information, contrary to position-based or distance-based formation control techniques. In addition, we do not need a global reference frame for the robots (which typically requires additional sensors). All the information is measured in the images, the cameras do not share a common coordinate reference, and their motion does not affect the control performance.

A preliminary version of the method proposed in this paper was presented in [19]. With respect to this previous work, we provide here several new contributions:

1) A more precise definition of the control method when multiple cameras are used. We now discuss specifically aspects of the control units that carry the cameras (UAVs), and define the interactions (established via communications) that need to exist between them to guarantee a correct performance of the global system.

2) A formal stability analysis of the multi-camera control method.

3) In addition to controlling the motion of the robots, here we also address the control of the motion of the cameras. We propose an algorithm that ensures that they maintain the required intersections between their sets of viewed ground robots and thus the desired multirobot task is successfully carried out.

4) The validation of the method via realistic simulations and experiments with multiple cameras and real mobile robots.

M. Aranda, G. López-Nicolás and C. Sagüés are with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain. {marandac, gonlopez, csagues}@unizar.es

Y. Mezouar is with Clermont Université, IFMA, Institut Pascal, Clermont-Ferrand, France. youcef.mezouar@ifma.fr

This work was supported by Ministerio de Ciencia e Innovación/European Union (project DPI2012-32100), Ministerio de Educación under FPU grant AP2009-3430, RobotEx Equipment of Excellence (ANR-10-EQPX-44) and the LabEx IMobS3 (ANR-10-LABX-16-01).

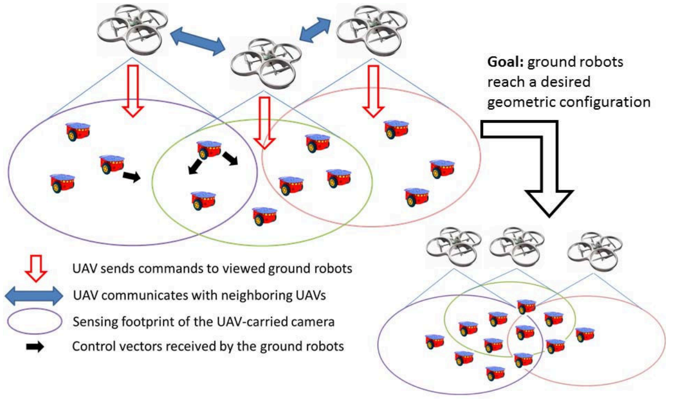
Goal: ground robots reach a desired geometric configuration
UAV sends commands to viewed ground robots
UAV communicates with neighboring UAVs
Sensing footprint of the UAV-carried camera
Control vectors received by the ground robots

Fig. 1. Overview of the proposed multirobot control system.

II. Image-based formation control framework

This section describes the framework the UAV units use to control the motion of the ground robots, relying on image information. Let us define next the elements of our system, which is illustrated in Fig. 1. We consider a set, S, of n unicycle robots and a set of m UAV control units, whose motion is modeled by single-integrator kinematics. Each unit carries a camera that views a subset of robots $S_j \subseteq S$, $j = 1, ..., m$. We denote $N_j = card(S_j)$. From the captured images, and using the proposed homography-based approach defined later on in this section, each unit obtains the motion commands for the robots within its camera's field of view. These take the form of desired motion vectors, and are transmitted to the robots. The robots that are viewed by multiple cameras combine the multiple received commands to obtain their motion input, as described in Section III. The UAVs also communicate among themselves in order to control their own motion (Section IV), with the goal of ensuring appropriate coverage of the robotic team and the successful completion of the control task, which consists in making the ground robots' positions form a desired shape, up to translation and rotation.

Let us focus next on a given unit j. We use normalized homogeneous image coordinates. Our formation control strategy uses two perspective images:

• The *reference image*, which is fixed and represents the desired configuration as viewed by a downward-facing camera. This can be an actual image captured while the robots were in that configuration, or a virtual image generated using the geometric constraints that define the formation, expressed in the image. Each robot $i \in S_j$ is represented by a point $\mathbf{p_i'}$, in pixel coordinates, in the reference image. All units have the same reference image of the full formation, but only use the points corresponding to robots they see.

• The *current image*, which represents the current configuration of the robots seen by the camera. This image is required to satisfy the following hypotheses: the image plane is parallel to the ground plane, and the distances, in pixels, in the current and reference images have the same scale. In the current image, each robot $i \in S_j$ is represented by a point, $\mathbf{p_i^j}$, in pixel coordinates.

If the two images are taken with the same camera and the motion of the sensor is planar, the two hypotheses are automatically satisfied. Let us stress, however, that the *actual* motion of the perspective camera in our method can be arbitrary. All that is required is a way to transform each image captured by the camera to an equivalent *current image* satisfying the assumptions above. In our previous work [16] we described in detail how this can be achieved through a purely image-based procedure that rectifies the captured images using a concatenation of transformations. This method compensates any 3D translational and rotational motion of the camera, and any scale differences between the reference and captured images.

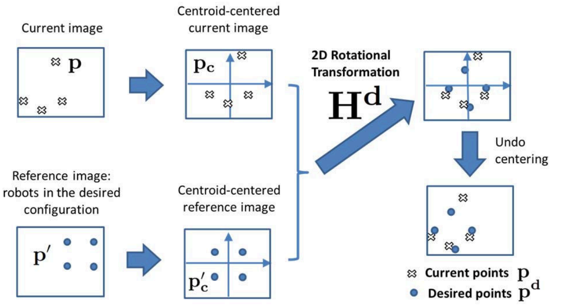
Current image
p
Centroid-centered current image
p_c
2D Rotational Transformation
H^d
Undo centering
Reference image: robots in the desired configuration
p′
Centroid-centered reference image
p′_c
Current points p
Desired points p^d

Fig. 2. Overview of the vision-based framework implemented by each of the control units using its associated camera. The computed desired image positions $\mathbf{p^d}$ for the robots are such that the sum of their squared distances to the current positions $\mathbf{p}$ is minimum, as explained in the text. The image-based control scheme we propose is based on driving $\mathbf{p}$ to $\mathbf{p^d}$.

In our scheme, each camera uses the points in the reference and current images to compute a *2D rigid transformation* composed of translation and rotation. Specifically, considering $i \in S_j$, when this transformation is applied to the reference points $\mathbf{p_i'}$, these are converted to a set of *desired points* in the current image, $\mathbf{p_i^{d,j}}$, such that the sum of squared distances between $\mathbf{p_i^j}$ and $\mathbf{p_i^{d,j}}$ is minimum. The solution to this type of rigid shape transformation problem [20] requires the coordinates of the points to be firstly translated so that they are centered on their centroids. Such centering results in two new sets of points $\mathbf{p_{ic}^j}$ and $\mathbf{p_{ic}'}$, $i \in S_j$. Then, a least-squares 2D rotation between them is obtained. As shown in [19], it is possible to express it in terms of linear image transformations, by computing from the two sets a similarity parameterized in the following way:

$$\mathbf{H_l^j} \sim \begin{bmatrix} s_j \cos\theta_j & s_j \sin\theta_j & 0 \\ -s_j \sin\theta_j & s_j \cos\theta_j & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

Then, the solution is given by the matrix $\mathbf{H^{d,j}} = \mathbf{H_l^j} \cdot diag(1/s_j, 1/s_j, 1)$, which is a constrained homography expressing a pure 2D rotation. The desired points in the current image of the camera can thus be defined, for $i \in S_j$, as:

$$\mathbf{p_i^{d,j}} = \mathbf{H^{d,j}} \mathbf{p_{ic}'} + \mathbf{c_p^j}, \tag{2}$$

where by summing the vector $\mathbf{c_p^j}$ (the centroid of the current points) we undo the previous centering. Figure 2 illustrates the geometry behind the process of determining the desired image points.

Considering the framework described, $\mathbf{p_i^j} = \mathbf{p_i^{d,j}}$ $\forall i \in S_j$ clearly implies the robots in this set are in the desired formation. Then, our control strategy relies on making them move to satisfy this condition. Our choice of a least-squares solution provides efficient task completion, since it minimizes the sum of squared distances from the robots' current positions to their destinations. Note the difficulty coming from the fact that each camera handles a *partial* set of robots. We need to define the intersections between these sets and the interactions between commands issued by different control units in a way that allows the *global* formation to be achieved. This is addressed in the following section.

III. Coordinated control of the ground robots

In this section, we employ the image-based framework presented in the previous section to propose a coordinated control scheme of the multirobot system employing multiple cameras. We explain next how a given control unit j computes the information to be sent to a

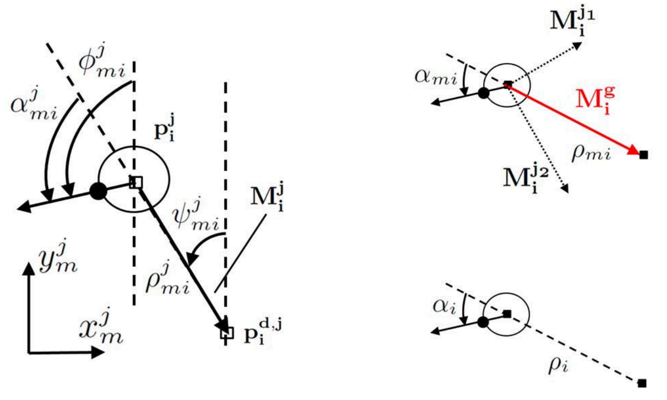
ϕ^j_{mi}
α^j_{mi}
$\mathbf{p}^{\mathbf{j}}_{\mathbf{i}}$
ψ^j_{mi}
$\mathbf{M}^{\mathbf{j}}_{\mathbf{i}}$
y^j_m
ρ^j_{mi}
x^j_m
$\mathbf{p}^{\mathbf{d,j}}_{\mathbf{i}}$
$\mathbf{M}^{\mathbf{j_1}}_{\mathbf{i}}$
α_{mi}
$\mathbf{M}^{\mathbf{g}}_{\mathbf{i}}$
ρ_{mi}
$\mathbf{M}^{\mathbf{j_2}}_{\mathbf{i}}$
α_i
ρ_i

Fig. 3. Left: geometric variables and control vector computed for robot i by unit j, defined in its image. Right-top: representation of $i's$ global motion vector computed from image information received from two units j_1 and j_2. Right-bottom: state of the robot on the ground plane.

given viewed robot $i \in S_j$. The parameters involved are illustrated in Fig. 3. Using the strategy described in the previous section, the unit can define $\mathbf{p}^{\mathbf{d,j}}_{\mathbf{i}}$ in the current image frame. The image distance between the current and desired points can be computed as:

$$\rho^j_{mi} = ||\mathbf{p}^{\mathbf{j}}_{\mathbf{i}} - \mathbf{p}^{\mathbf{d,j}}_{\mathbf{i}}|| = \sqrt{(p^j_{xi} - p^{d,j}_{xi})^2 + (p^j_{yi} - p^{d,j}_{yi})^2} \, . \quad (3)$$

We express all angles in $(-\pi, \pi]$. The angle ψ^j_{mi} can be obtained as:

$$\psi^j_{mi} = atan2\left(-(p^j_{xi} - p^{d,j}_{xi}), (p^j_{yi} - p^{d,j}_{yi})\right) \, , \quad (4)$$

while ϕ^j_{mi} is calculated directly from the image of the robot. The alignment error in the image, α^j_{mi}, is obtained as $\alpha^j_{mi} = \phi^j_{mi} - \psi^j_{mi}$. From ρ^j_{mi} and α^j_{mi}, the unit can express $\mathbf{p}^{\mathbf{d,j}}_{\mathbf{i}}$ and the resulting motion vector $\mathbf{M}^{\mathbf{j}}_{\mathbf{i}} = \mathbf{p}^{\mathbf{d,j}}_{\mathbf{i}} - \mathbf{p}^{\mathbf{j}}_{\mathbf{i}}$, in the robot's frame. Then, this vector is what the unit transmits to the robot.

Robot i may receive this control information from multiple units at any time. When this is the case, it sums the individual vectors to obtain its global motion vector: $\mathbf{M}^{\mathbf{g}}_{\mathbf{i}} = \sum_{j \in C_i} \mathbf{M}^{\mathbf{j}}_{\mathbf{i}}$, where C_i is the set of indexes of the control units that send control data to robot i. This strategy to integrate the information seems intuitively reasonable since the robot's motion averages the motions commanded by the units. Referring to Fig. 3, the robot can then directly compute the parameters α_{mi} and ρ_{mi}. For the purposes of our control strategy, we can simply express the state of the robot via the position on the ground associated with the endpoint of its global motion vector. We can express this position relative to the robot by the variables α_i and ρ_i. Note that $\alpha_i = \alpha_{mi}$ and ρ_i is proportional to ρ_{mi}. This permits us to control the robot using the quantities measured in the image. In particular, we consider the robots have unicycle kinematics. Then, the control law we propose for robot i is:

$$\left\{ \begin{array}{l} v_i = -k_v \, sign(\cos\alpha_{mi}) \, \rho_{mi} \\ \omega_i = k_\omega \, (\alpha_{di} - \alpha_{mi}) \end{array} \right. , \quad (5)$$

where ω_i is defined counterclockwise, $k_v > 0$ and $k_\omega > 0$ are control gains, and we define:

$$\alpha_{di} = \left\{ \begin{array}{lll} 0 & \text{if} & |\alpha_{mi}| \leq \frac{\pi}{2} \\ \pi & \text{if} & |\alpha_{mi}| > \frac{\pi}{2} \end{array} \right. .$$

With this control, each robot can travel forwards or backwards depending on its location relative to its motion vector. Observe that if $\cos\alpha_{mi} = 0$ for some robot, the agent does not translate (since $v_i = 0$), and can only rotate in place. Note as well that the final heading of each robot is not controlled to a particular value. If required, the unicycle robots can achieve any desired heading by rotating in place once they have reached their final positions.

A. Stability analysis

When considering a single camera, the proposed homography-based control framework is asymptotically stable, with locally exponential convergence, as analyzed in [16], [19]. Particularly, in [19] and in the present work we use a purely rotational transformation matrix ($\mathbf{H}^{\mathbf{d,j}}$) not encoding any translation, which simplifies the stability analysis with respect to the system in [16]. We focus here on the multi-camera control law, and analyze its stability next.

Note that, clearly, the desired points obtained as described in Section II are independent of the image frame used to compute them. Thus, for our analysis we consider henceforth that all the image entities used by the different cameras are expressed in a *common* image frame. Let us define an undirected graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$ modeling the intersections between the sets S_j, $j = 1, ..., m$, which are assumed fixed. In particular, every node in $\mathcal{V}_c$ corresponds with one camera, and there is a link in $\mathcal{E}_c$ between two nodes j and k when it holds that $card(S_j \bigcap S_k) \geq 2$. We assume the graph $\mathcal{G}_c$ is connected, and every robot is viewed by at least one camera, i.e. $\bigcup_{j=1,...,m} S_j = S$. In addition, we further assume the transformations $\mathbf{H}^{\mathbf{d,j}}$ for $j = 1, ..., m$ are non-degenerate. Then, we obtain the following result:

Proposition 1: The multirobot system under the control law (5) with multiple cameras is locally stable with respect to the desired formation.

Proof: We will use Lyapunov analysis to demonstrate the stability of the system. Let us define the following cost function for every control unit j:

$$V^j = \frac{1}{2}\sum_{i \in S_j} ||\mathbf{p}^{\mathbf{d,j}}_{\mathbf{i}} - \mathbf{p_i}||^2 = \frac{1}{2}\sum_{i \in S_j} ||\mathbf{H}^{\mathbf{d,j}}\mathbf{p}'_{\mathbf{ic}} + \mathbf{c}^{\mathbf{j}}_{\mathbf{p}} - \mathbf{p_i}||^2, \quad (6)$$

where (2) has been used. Note that $\mathbf{H}^{\mathbf{d,j}}$ is the 2D rotational homography computed with the robots in S_j, and $\mathbf{c}^{\mathbf{j}}_{\mathbf{p}} = (\sum_{i \in S_j} \mathbf{p_i})/N_j$. Then, we define the following candidate Lyapunov function:

$$V = \sum_{j=1,...,m} V^j. \quad (7)$$

Note that V is positive semi-definite and radially unbounded. Furthermore, it can be easily seen that, thanks to $\mathcal{G}_c$ (which links the cameras that share at least two robots) being connected, $V = 0$ occurs if and only if the n robots are in the desired configuration. We address next the study of the dynamics of V. From (6) and (7), we can write:

$$\dot{V} = \sum_{j=1,...,m} \left[\sum_{i \in S_j} \left(\frac{\partial V^j}{\partial \mathbf{p_i}} \right)^T \dot{\mathbf{p}}_{\mathbf{i}} \right] . \quad (8)$$

Observe that from (6), for any $i \in S_j$ we can write:

$$\frac{\partial V^j}{\partial \mathbf{p_i}} = \frac{\partial V^j}{\partial \mathbf{H}^{\mathbf{d,j}}} \frac{\partial \mathbf{H}^{\mathbf{d,j}}}{\partial \mathbf{p_i}} + \frac{1}{N_j}\sum_{k \in S_j} (\mathbf{p}^{\mathbf{d,j}}_{\mathbf{k}} - \mathbf{p_k}) + (\mathbf{p_i} - \mathbf{p}^{\mathbf{d,j}}_{\mathbf{i}}), \quad (9)$$

considering the non-degenerate least-squares rotation $\mathbf{H}^{\mathbf{d,j}}$ is a differentiable function of the image points [20]. Since this homography is computed in our method so as to minimize the sum of squared distances expressed in the cost function V^j, we have that $\frac{\partial V^j}{\partial \mathbf{H}^{\mathbf{d,j}}}$ is null. In addition, as can be deduced from Section II, the centroids of the current and desired points in S_j coincide. Therefore, the second addend on the right-hand side of (9) is also null and we get: $\frac{\partial V^j}{\partial \mathbf{p_i}} = \mathbf{p_i} - \mathbf{p}^{\mathbf{d,j}}_{\mathbf{i}}$. We can then express (8) as follows:

$$\dot{V} = \sum_{j=1\ldots m}\sum_{i\in S_j}(\mathbf{p_i}-\mathbf{p_i^{d,j}})^T\dot{\mathbf{p}}_\mathbf{i} = \sum_{i=1\ldots n}\sum_{j\in C_i}(\mathbf{p_i}-\mathbf{p_i^{d,j}})^T\dot{\mathbf{p}}_\mathbf{i}. \quad (10)$$

Robot $i's$ global vector (Section III) in the common frame is:

$$\mathbf{M_i^g} = \sum_{j\in C_i}\mathbf{M_i^j} = \sum_{j\in C_i}(\mathbf{p_i^{d,j}} - \mathbf{p_i}). \quad (11)$$

Let us consider first that the robots are holonomic (i.e. they can travel in any spatial direction in the plane). If that were the case, robot i would move in the direction of the vector, and thus the dynamics of its associated image point would be: $\dot{\mathbf{p}}_\mathbf{i} = k_v'\mathbf{M_i^g}$ for some $k_v' > 0$. Therefore, from (10) and (11):

$$\dot{V} = -k_v'\sum_{i=1,\ldots,n}||\mathbf{M_i^g}||^2 \leq 0. \quad (12)$$

For the type of robots we consider in this work (i.e. unicycles), the translation is always in the direction of the robot's current heading, and the magnitude of the motion is proportional to that of the global motion vector (5). The misalignment between the image projection ($\dot{\mathbf{p}}_\mathbf{i}$) of the actual translation vector and the direction of the global motion vector is expressed by the angle $\alpha_{di} - \alpha_{mi}$. Notice that (10) captures the dot product of these two vectors. We can thus write:

$$\dot{V} = -k_v'\sum_{i=1,\ldots,n}||\mathbf{M_i^g}||^2\cos(\alpha_{di} - \alpha_{mi}). \quad (13)$$

Observe that $0 \leq |\alpha_{di} - \alpha_{mi}| \leq \pi/2$ and, therefore, it holds as well that $\dot{V} \leq 0$ for unicycle kinematics. Then, by virtue of the global invariant set theorem, we can conclude that the system converges asymptotically to the largest invariant set in the set $R = \{\mathbf{p_i}, i = 1, ..., n \mid \dot{V} = 0\}$, and the system is locally stable with respect to the desired equilibrium $V = 0$, i.e. with respect to the desired formation. ■

Corollary 1: Assume that $\mathcal{G}_c$ is a tree, $card(S_i \bigcap S_j) = 2$ if $\{i, j\} \in \mathcal{E}_c$, 0 otherwise, and $S_i \bigcap S_j \bigcap S_k = \emptyset$ for unequal i, j, k. Then, the system converges globally to the desired formation.

Proof: We want to determine the conditions under which $\dot{V} = 0$ can hold. Due to the unicycle kinematics, $\dot{V}$ can be zero when, with no robot translating, at least one of them satisfies $\cos(\alpha_{di} - \alpha_{mi}) = 0$ while $||\mathbf{M_i^g}|| > 0$ (13). From (5), these robots will rotate in place at that instant, making $\dot{V} < 0$. Thus, we only need to study the case where $\dot{V} = 0$ due to $\mathbf{M_i^g} = \mathbf{0}$ $\forall i$. Assume this scenario, which implies all the robots are static, holds at some instant. Consider a leaf node j_1 in $\mathcal{G}_c$ connected with a node j_2, and denote $S_{j_1} \bigcap S_{j_2} = \{i_1, i_2\}$.

Notice that since the $N_{j_1} - 2$ robots controlled *only* by j_1 are static, they already are in their desired positions computed by j_1, and the only possible nonzero motion vectors computed by this camera are for robots i_1 and i_2. The centroids of the current and desired points for any control unit coincide (Section II), and therefore $\mathbf{M_{i_1}^{j_1}} = -\mathbf{M_{i_2}^{j_1}}$. Given that i_1 and i_2 are also static (i.e. $\mathbf{M_i^g} = \mathbf{0}$ for them) we also have $\mathbf{M_i^{j_2}} = -\mathbf{M_i^{j_1}}$ for $i = i_1, i_2$. We consider now an orthogonality condition for the least-squares rigid shape alignment problem we treat [20]. In particular, let us assume, without loss of generality, that the image points are expressed in a coordinate frame such that the centroid of the points in camera j_1 is null and $\mathbf{H^{d,j_1}} = \mathbf{I_3}$. Then, it holds that $\sum_{i=1,\ldots,N_{j_1}}(\mathbf{p_i^{d,j_1}})^T\mathbf{p_i^\perp} = 0$, where $\perp$ indicates a rotation of $\pi/2$ radians. As the current and desired positions coincide for all robots in S_{j_1} except i_1 and i_2, it is straightforward to see from the above expressions that $(\mathbf{p_{i_1}^{d,j_1}} - \mathbf{p_{i_2}^{d,j_1}})^T(\mathbf{p_{i_1}} - \mathbf{p_{i_2}})^\perp = 0$. Together with the previously established conditions, this clearly implies that *all* motion vectors for the two robots from the two cameras must lie along the line that joins $\mathbf{p_{i_1}}$ and $\mathbf{p_{i_2}}$. Figure 4 illustrates the image points and motion vectors resulting from all the described constraints.

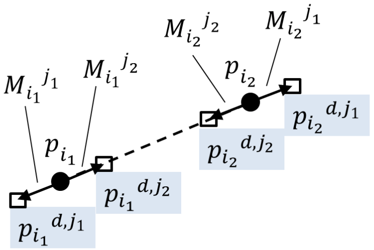

Fig. 4. Image points and motion vectors for two robots i_1 and i_2 controlled by two control units j_1 and j_2 under the conditions of Corollary 1.

Assume now, without loss of generality, that $||\mathbf{p_{i_1}^{d,j_1}} - \mathbf{p_{i_2}^{d,j_1}}|| > ||\mathbf{p_{i_1}} - \mathbf{p_{i_2}}||$. This implies $||\mathbf{p_{i_1}^{d,j_2}} - \mathbf{p_{i_2}^{d,j_2}}|| < ||\mathbf{p_{i_1}} - \mathbf{p_{i_2}}||$. However, since the distance between desired points must clearly be equal for j_1 and j_2, we conclude that $\mathbf{M_i^j} = \mathbf{0}$ for $i = i_1, i_2$ and $j = j_1, j_2$. Then, using that the intersections between S_j sets contain two robots and are mutually disjoint, it is straightforward to propagate this reasoning from the leaf nodes through the full tree graph and obtain that $\mathbf{M_i^j} = \mathbf{0}$ $\forall i, j$, i.e. $V = 0$. Thus, $\dot{V} = 0 \Leftrightarrow V = 0$, which implies the system converges globally to the desired formation. ■

Remark 1: Global stability guarantees such as the ones we have provided are well known to be difficult to obtain for non-centralized formation stabilization in the absence of a global reference frame [4], [5]. In our simulations, fairly diverse setups (in terms of numbers of robots and cameras, robot intersections, and topologies of $\mathcal{G}_c$) have been tested, and we have not observed local minima (i.e. the desired formation was always achieved). Notice, in any case, that the presence of the control units as supervisory elements provides flexibility to escape the local minima, in case they occurred. This can be achieved by switching to the provably globally stable system setup defined in Corollary 1. Observe that this switch can be performed simply by making the aerial units control only some of the robots in their field of view, and does not require the UAVs to move.

Remark 2: Consider the case where the sets S_j are time-varying, i.e. there are ground robots entering or leaving the cameras' fields of view during the execution of the control. Then, when for some camera j one robot is added to S_j, this may increase V^j instantaneously and can cause the aggregate function V to be non-decreasing at that instant. Observe that these additions will depend on the specific policy used to control the motion of the cameras and regulate the changes in the S_j sets. If the policy guarantees that the number of robot additions is finite over time, the graph $\mathcal{G}_c$ will eventually become static and therefore the stability of the system will be guaranteed.

Remark 3: As mentioned before the statement of Proposition 1, we consider the rigid transformations used in our method do not suffer from degeneracies, which can appear e.g. if multiple robots occupy the same position, and their image projections coincide. The configurations causing these degeneracies have measure zero, i.e. will never occur in reality. Therefore, our system is globally stable in practice, as per Corollary 1. We note that, theoretically, if these configurations were considered, only *almost global* stability would be guaranteed for the system under our control strategy.

IV. Camera motion control

In this section, we propose an algorithm to control the motion of the UAVs carrying the cameras. We assume that the cameras are downward-facing and the UAVs have single integrator kinematics. We consider that two UAVs whose fields of view intersect can communicate. Specifically, they exchange the information of which robots they are viewing. We also assume that the conditions discussed in the previous section (i.e. all the ground robots are covered, and the graph $\mathcal{G}_c$ must be connected) hold at the start of the execution. Algorithm 1 outlines a policy that effectively guarantees these requirements are always met, thus ensuring convergence of the formation.

Algorithm 1 Motion of the camera/control unit j

```
1) For every control unit $k$ that is a neighbor in the defined initial graph $\mathcal{G}_{c0}$, $j$ determines through communications the set $S_{jk} = S_j \bigcap S_k$, and the set of ground robots which are only viewed by its camera, $S_{jj}$.
2) From its camera's image information, $j$ determines, for every neighboring $k$, which two robots in $S_{jk}$ are closest to itself: $S_{jkc} = \{r_{jk1}, r_{jk2}\}$. Then, it determines the robot $r_{jf}$ in the set $S_{jc} = \bigcup_k \{S_{jkc}\}$ that is farthest away from $j$.
3) Unit $j$ computes in its camera's image the centroid, $\mathbf{c_h}$, of the set of points $S_{jc} \bigcup S_{jj}$.
4) Unit $j$ computes its motion command. Its horizontal velocity is toward $\mathbf{c_h}$, proportional to the distance to that point (measured in $j's$ camera images). Vertically, it moves upwards if the distance to $r_{jf}$ in the images is greater than a predefined desired distance (i.e. when $r_{jf}$ is near the border of $j's$ camera's field of view), and downwards if it is smaller. The vertical velocity is proportional to the difference between the current and desired image distances for $r_{jf}$.
5) Unit $j$ executes its own motion and transmits the control commands to the robots in $S_j$.
```

This method maintains the links of a fixed connected graph defined initially, $\mathcal{G}_{c0}$. The main purpose of controlling the camera height is to ensure the necessary robots stay in the field of view. In addition, the UAVs will move downwards, when possible, to get a higher resolution view of the robots. A minimum height must be defined, for safety. The horizontal motion strategy aims at maintaining good visibility of the robots that the camera has to *preserve* within its field of view. If multiple UAVs detect that they are preserving the same set of robots, all but one of them must be stopped, to avoid collisions. Every ground robot will remain covered as long as it does not leave the field of view of multiple cameras simultaneously. This can be easily avoided using safety margins and appropriate selection of the control gains for the robots and UAVs.

V. Simulations

In this section, we describe simulation results to evaluate the performance of the proposed control scheme. We present first a simulation carried out using Matlab, aimed at illustrating the scalability of our method. Five UAVs and forty unicycle robots were used, with a rectangular grid-shaped desired configuration. The cameras' fields of view were approximated by circles, and the UAV motions were controlled using the algorithm outlined in Section IV. A cycle graph $\mathcal{G}_{c0}$ was used. Notice in Fig. 5 how the target configuration is achieved while the cameras maintain the group coverage (as illustrated by their sensing footprints). As expected (Section III-A), instantaneous jumps in the cost functions appear when new robots enter the sets S_j. Still, this effect does not compromise the stability of the system.

We also tested our approach using the Cobaye software package developed by the company 4D-Virtualiz[1]. This is a realistic simulator of mobile robotic systems which includes the modeling of dynamics and permits real-time operation. We illustrate a simulation example in an urban scenario with twelve Pioneer 3-AT robots and three UAVs, each carrying a downward-facing perspective camera. The size of the images was 800 × 800, and the cameras' field-of-view half-angle was 35°. The desired configuration for the multirobot team had the shape of a circle. Initially, the robots had arbitrary positions and orientations, with the three UAVs covering them. We defined $\mathcal{G}_{c0}$ to be a linear graph. The results are illustrated in Figs. 6 and 7. The robots converged to the desired configuration following fairly smooth trajectories, while the UAVs jointly maintained visual coverage of the

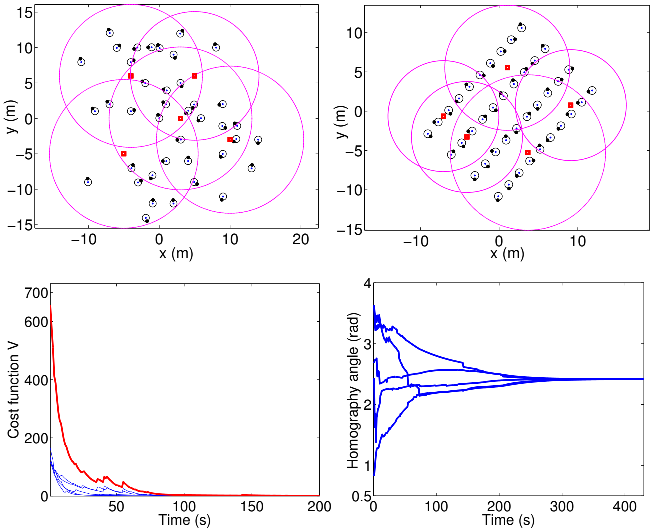

Fig. 5. Simulation results with forty robots and five cameras. Top: Initial (left) and final (right) configurations, showing the robots (circles), the cameras (squares), and the circular footprints associated with their field-of-view. Bottom: cost functions for every camera, and Lyapunov function, shown in a thicker line (left). Angles of the 2D rotational homographies computed by every camera, in a common reference frame (right).

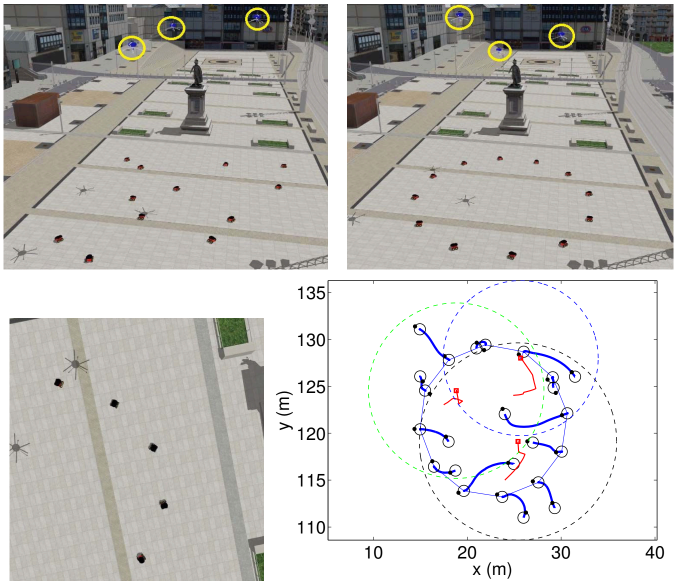

Fig. 6. Results from the realistic simulation example. Top: Initial (left) and final (right) views of the simulated setting, with circles plotted around the UAVs. Bottom: Final image captured by one of the cameras (left). Paths followed by the robots (right). The cameras' paths are displayed in thin lines, and their final positions are marked with squares. The final camera footprints are shown as dashed-line circles.

[1] www.4d-virtualiz.com

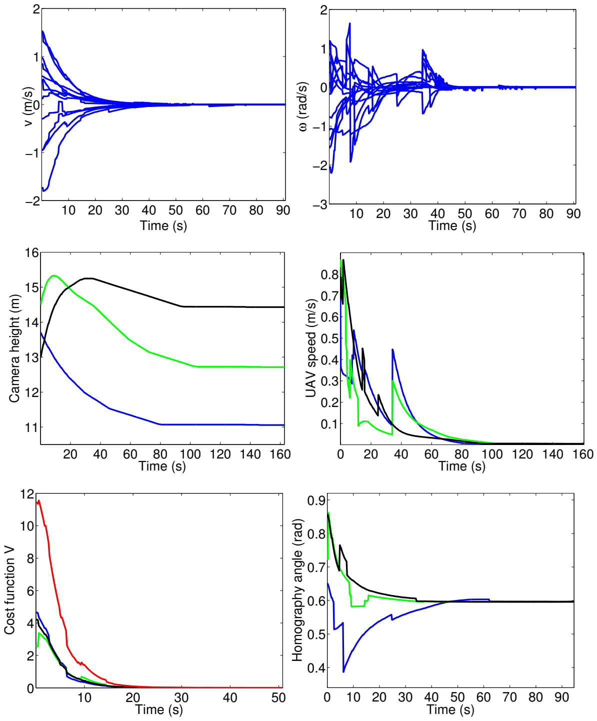

Fig. 7. Results from the realistic simulation. Top: Linear (left) and angular (right) velocities followed by the robots. Middle row: Evolution of the camera heights (left) and magnitudes of the UAV velocities (right). Bottom: cost functions for the cameras (the curve having greater values is the global Lyapunov function) (left), and angles of the cameras' 2D rotational homographies, expressed in a common frame (right).

team, and their 3D positions eventually stabilized. The effects of the changes in the S_j sets can be observed in the displayed plots.

VI. Experiments with Real Robots

We tested our control method using four Khepera III robots moving on a planar workspace. Different FireWire cameras observing the robots were used to obtain the results we present next. Circular-coded patterns were placed on top of the robots to allow them to be detected and identified in the images. Four additional markers were placed on the planar workspace, for image rectification purposes. Specifically, the points in the captured images were transformed to enforce the hypotheses we require for the current image (Section II). The details of the rectification procedure employed can be found in [16]. A set of image positions obtained with the robots forming the desired configuration was used as the reference image for the homography-based control computations. The distance between the fixed markers in the reference image was used to fix the scale of the current image. Thus, no metric or camera calibration information was employed in the experiments.

We describe the results of an experiment carried out with two cameras. One of them was hand-held during the experiment, performing a motion comprising both translation and rotation. This camera was equipped with a lens having a focal length of 5 mm. The other camera had a lens of 3.6 mm focal length and was fixed over the robots' workspace, facing downward. Both cameras viewed all four robots, but we defined (naming the cameras 1 and 2 and the robots r_1, r_2, r_3, r_4) $S_1 = \{r_1, r_2, r_3\}$ and $S_2 = \{r_2, r_3, r_4\}$, so as to test the performance of the proposed distributed controller. The desired configuration was square-shaped, and the control loop ran at 5 frames/s. Figure 8 shows the time evolution of the velocities sent to the robots for this experiment, computed according to the method described in Section III. The evolutions of the angles of the 2D rotational homography transformations computed by the two cameras are also shown. For the fixed camera, the computed angle eventually stabilizes, whereas for the hand-held one it keeps changing over time, due to the camera motion. The same figure displays the cost functions (as defined in Section III-A) for the two cameras, and the Lyapunov function. They all vanish as the robots converge to the desired formation. We show in Fig. 9 images acquired by the cameras, overlaying the traces of the robots as the control is executed. For the fixed camera, the image paths of the robots illustrate what their real paths were. The effects of the motion of the hand-held camera are apparent in its corresponding robot traces. We also show a sequence of images from a different two-camera experiment, this time with a T-shaped desired configuration, to further illustrate the image motion taking place while the control is running. Supplementary illustration of the simulation and experimental results is provided by the videos that accompany the paper.

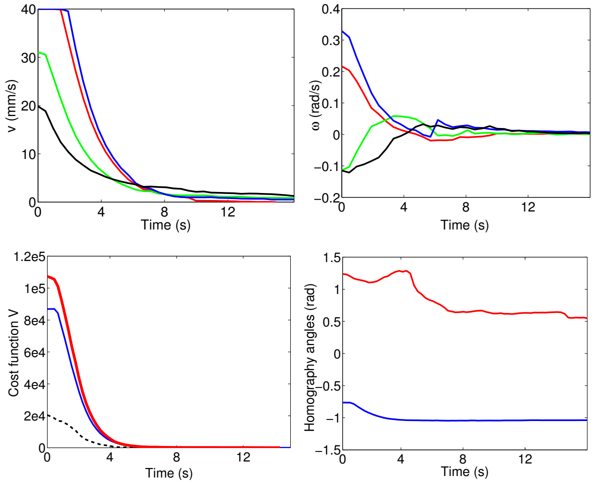

Fig. 8. Results from the experiment with real robots. Top row: Linear (left) and angular (right) velocities sent to the four robots. Bottom row: cost functions for the two cameras (the dashed line corresponds to the moving camera), and global Lyapunov function V (thicker line) (left). Evolution of the angle of the 2D rotational homography computed by the two cameras. The positive-valued curve corresponds to the moving camera (right).

VII. Conclusion

We presented a homography-based multirobot control approach where multiple UAVs carrying cameras observe and control a set of robots moving on the ground plane to bring them to a desired formation. The proposed partially distributed controller is robust and scalable with respect to the number of robots and the size of the formation. Its effectiveness was validated in simulations and experiments with real robots.

References

[1] M. M. Zavlanos and G. J. Pappas, "Distributed formation control with permutation symmetries," in *IEEE Conference on Decision and Control*, 2007, pp. 2894–2899.

[2] L. Sabattini, C. Secchi, and C. Fantuzzi, "Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation," *Autonomous Robots*, vol. 30, no. 4, pp. 385–397, 2011.

[3] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[4] D. V. Dimarogonas and K. H. Johansson, "Further results on the stability of distance-based multi-robot formations," in *American Control Conference*, 2009, pp. 2972–2977.

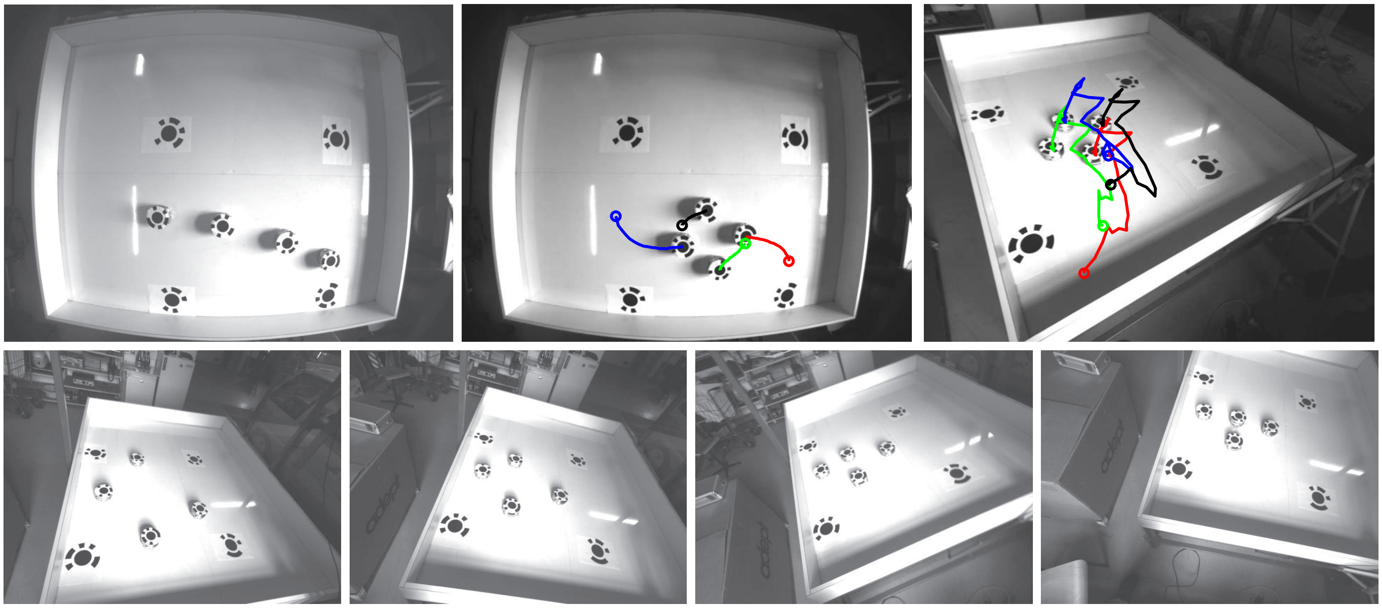

Fig. 9. Results from real experiments. The desired formation has the shape of a square (top row) and a ”T” (bottom row). Top row: Initial image from the fixed camera (left). Final image from the fixed camera, with the image traces of the robots during the control execution overlaid (center). Final image, with the robots’ image traces overlaid, for the moving camera (right). Bottom row: sequence of images from another experiment. The initial image (left), the final one (right) and two intermediate images from the moving camera are displayed.

[5] J. Guo, Z. Lin, M. Cao, and G. Yan, “Adaptive control schemes for mobile robot formations with triangularised structures,” *IET Control Theory & Applications*, vol. 4, no. 9, pp. 1817–1827, 2010.

[6] M. Ji and M. Egerstedt, “Distributed coordination control of multiagent systems while preserving connectedness,” *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[7] D. V. Dimarogonas and K. J. Kyriakopoulos, “A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems,” *Automatica*, vol. 44, no. 10, pp. 2648–2654, 2008.

[8] B. S. Park, J. B. Park, and Y. H. Choi, “Adaptive formation control of electrically driven nonholonomic mobile robots with limited information,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 4, pp. 1061–1075, 2011.

[9] J. Yu, S. M. LaValle, and D. Liberzon, “Rendezvous without coordinates,” *IEEE Trans. Automat. Contr.*, vol. 57, no. 2, pp. 421–434, 2012.

[10] G. López-Nicolás and Y. Mezouar, “Special issue on visual control of mobile robots,” *Robotics and Autonomous Systems*, vol. 62, no. 11, pp. 1611–1612, 2014.

[11] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, “A vision-based formation control framework,” *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2002.

[12] R. Vidal, O. Shakernia, and S. Sastry, “Following the flock: Distributed formation control with omnidirectional vision-based motion segmentation and visual servoing,” *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 14–20, 2004.

[13] N. Moshtagh, N. Michael, A. Jadbabaie, and K. Daniilidis, “Vision-based, distributed control laws for motion coordination of nonholonomic robots,” *IEEE Trans. on Robotics*, vol. 25, no. 4, pp. 851–860, 2009.

[14] D. Panagou and V. Kumar, “Cooperative visibility maintenance for leader-follower formations in obstacle environments,” *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 831–844, 2014.

[15] N. Michael, J. Fink, and V. Kumar, “Controlling ensembles of robots via a supervisory aerial robot,” *Advanced Robotics*, vol. 22, no. 12, pp. 1361–1377, 2008.

[16] G. López-Nicolás, M. Aranda, Y. Mezouar, and C. Sagüés, “Visual control for multirobot organized rendezvous,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 4, pp. 1155 –1168, 2012.

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

[18] X. C. Ding, A. Rahmani, and M. Egerstedt, “Multi-UAV convoy protection: An optimal approach to path planning and coordination,” *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 256–268, 2010.

[19] M. Aranda, Y. Mezouar, G. López-Nicolás, and C. Sagüés, “Partially distributed multirobot control with multiple cameras,” in *American Control Conference*, 2013, pp. 6308–6314.

[20] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*. Oxford University Press, 2004.