Citation: M. Aranda, G. López-Nicolás, and C. Sagüés. Controlling multiple robots through multiple 1D homographies. IEEE International Conference on Systems, Man, and Cybernetics (SMC 2013), pp. 589-594. Manchester, UK, October 13-16, 2013

Controlling Multiple Robots through Multiple 1D Homographies

Miguel Aranda, Gonzalo López-Nicolás and Carlos Sagüés
Instituto de Investigación en Ingeniería de Aragón
Universidad de Zaragoza, Spain
Email: {marandac,gonlopez,csagues}@unizar.es

***Abstract*—We present a method for visual control of a set of robots moving on the ground plane. The goal of the control task is for the team to reach a desired geometric configuration. Each robot carries an omnidirectional camera and can communicate with a number of the other robots. The approach relies on the computation of the planar motion between two views, by means of 1D homographies. This knowledge, obtained by each robot from its own images and the visual information received from neighboring robots, allows it to define a desired position on the plane. Then, we propose a novel control scheme based on computing a particular 2D transformation to drive each robot towards its goal position. A contribution of this work is the use of 1D homography in a multirobot control framework. This tool allows to deal with purely angular visual information, which is precise and requires no calibration. The approach we present is completely distributed. Each robot uses only information from its formation neighbors and the global centroid to obtain its motion commands. These individual behaviors naturally result in the complete team of robots reaching the desired global configuration.**

Keywords*—*multirobot systems; vision-based control; mobile robots; multiple-view geometry; formation control

I. Introduction

Research in the field of multirobot systems has grown in recent years. A variety of tasks in robotics can be carried out more efficiently with a group of agents than with a single one. Moreover, technological advances are making the implementation of systems featuring multiple robots more and more feasible every day. The problem of controlling the motion of multiple robots to enable the execution of a desired task finds applications in such areas as cooperative exploration, perception, search and rescue or surveillance.

We address in this work the control of a set of robots moving on a planar ground so that the group attains a desired geometric configuration. Many works have tackled the problem of enabling teams of ground robots to reach and maintain a formation shape [1]–[6]. The specific characteristics of the sensing and the communications within the group are key to this task. With regard to the sensing, we use vision, which provides abundant information at a relatively modest cost. Vision sensors have been employed previously in a number of related works in the context of multirobot control, including both centralized [7], [8] and distributed [9], [10] schemes. These two categories of approaches to the design of systems consisting of multiple robots have been extensively explored in the literature, although it appears that distributed methods are becoming increasingly popular due to their robustness, scalability and flexibility.

A very relevant difference between this paper and the existing works on vision-based multirobot control is given by our use of the 1D homography model [11]–[13]. This allows us to exploit the planar camera motion constraint and benefit from the robustness typically associated to multiple-view geometric models [11]. We use omnidirectional cameras, a natural choice given that we work with purely angular information. The available amount of angular information is maximized when cameras of this type are employed, thanks to their large field of view. In our approach, each robot is assumed to carry an omnidirectional camera. By using its own images and the visual information received from other agents, the robot obtains an estimation of its relative position with respect to the others. This task is carried out employing the epipoles computed from the 1D homographies, and a state observer which calculates the distances to the other robots. Using this information, we define a distributed control scheme which drives the robot team to the desired configuration. In particular, the scheme is based on geometric transformations computed by each robot using the relative positions of its formation neighbors.

Let us highlight a number of advantages of the proposed method: first, the angular visual information provided by omnidirectional vision is rich and precise, and it can be extracted without the need for specific calibration. In fact, any sensor providing angular information can be used. When considering the problem of planar structure and motion computation from angular information, the 1D trifocal tensor is the standard geometric model available [14], [15]. However, we address this problem employing a two-view model (the 1D homography) which is more efficient to compute and simpler to use. Also, our technique relies only on the visual information extracted from the scene surrounding the set of robots. Thus, we do not require to visually perceive, segment and identify the neighbors, and the method is robust to their occlusion. Concerning the control scheme we propose, it has interesting properties: it is coordinate-free, completely distributed, and efficient to compute, and it generates satisfactory robot trajectories.

II. Planar motion from two 1D views

Our multirobot control method relies on the computation of an unconstrained planar (i.e. occurring in a 2D space) motion from two 1D views. Let us analyze the feasibility of this task and its information requirements. Consider the Euclidean

This work was supported by Ministerio de Economía y Competitividad/European Union (project DPI2012-32100), by Ministerio de Educación (FPU grant AP2009-3430), and by DGA-FSE (group T04).

structure and motion problem for a set of landmarks in 2D projected in two calibrated 1D views. The position of each landmark is determined with two parameters, and the relative motion is determined with three (one for the rotation and two for the translation). Since the reconstruction is obtained up to an overall scale factor, the number of parameters needed is reduced by one. As the projection of each landmark in the two 1D views provides two equations, we have, for n landmarks, $2n + 3 - 1$ parameters and $2n$ equations. Thus, there are always two more parameters than equations, and the problem is not solvable, regardless of the number of landmarks. It is well-known [14], [15] that three views are needed to solve the 2D structure and motion problem in a general case. However, we can try to consider restrictions in the landmark positions in the two-view case. Indeed, if three landmarks belong to the same line in the 2D space, there is one geometric constraint relating their coordinates. With two different lines, we have two independent constraints, and the problem becomes solvable. Thus, if we have landmarks belonging to two different lines (with at least three landmarks in each of the lines) we can compute the 2D motion from two 1D views.

What is important is that there exists a convenient way to exploit this possibility, using 1D homographies. This is discussed in the following section.

A. Motion from 1D homographies

A 1D homography is a projective transformation between two 1D views induced by a line in the 2D projective space. It is expressed as a 2×2 matrix with three degrees of freedom. It is known that from the 1D projections in two views of at least three landmarks belonging to a line in 2D, we can compute a 1D homography by solving a linear system. Thus, from a putative set of 1D correspondences, we can look for two different 1D homographies, each fitting a subset of the correspondences. Then, notice that these two homographies capture exactly the required information to compute the 2D motion, as discussed in the previous section. The advantage of using 1D homographies for the task is that they can be computed linearly and robustly. Moreover, as shown in our previous work [13], the motion can be extracted from the two 1D homographies in a straightforward manner. In the mentioned work, to which we refer the interested reader, the planar motion computation procedure is explained in detail. The basic aspects of this method are reviewed next.

The projections in two 1D images of points belonging to a line in a planar scene are related by a 1D homography. The motion between the images is encapsulated by this transformation, $\mathbf{H_c} \in \mathbb{R}^{2\times2}$, which can be expressed as a function of the angle of rotation, ϕ, the translation vector, $(t_1, t_2)^T$, the normal of the line, $(n_1, n_2)^T$, the distance to the line, d, and a real-valued scale factor λ:

$$\mathbf{H_c} = \lambda \begin{bmatrix} \cos\phi + t_1 n_1/d & -\sin\phi + t_1 n_2/d \\ \sin\phi + t_2 n_1/d & \cos\phi + t_2 n_2/d \end{bmatrix}. \tag{1}$$

As shown analytically in [13], the computation of the motion between images from $\mathbf{H_c}$ results in a family of infinite valid solutions. It is therefore not possible to compute the motion from one 1D homography. Then, we propose a method employing two homographies induced by two different lines. Assuming that two different 1D homographies, $\mathbf{H_1}$ and $\mathbf{H_2}$, can be computed between two views, it is possible to calculate two projective transformations that map the images to themselves, called 1D homologies. We can define these transformations, called $\mathbf{G_1}$ and $\mathbf{G_2}$, as follows:

$$\begin{aligned} \mathbf{G_1} &= \mathbf{H_2^{-1} H_1} \\ \mathbf{G_2} &= \mathbf{H_2 H_1^{-1}}, \end{aligned} \tag{2}$$

where $\mathbf{G_1}$ maps view 1 to itself and $\mathbf{G_2}$ maps view 2 to itself. Thanks to a well-known property of the homology, the epipoles between the two 1D views can be obtained as eigenvectors of the homology matrices $\mathbf{G_1}$ and $\mathbf{G_2}$. Then, if the angle of the epipole in view 1 is α_{12} and the angle of the epipole in view 2 is α_{21}, we can compute the relative motion (i.e. the rotation angle, ϕ, and the angle of the translation vector, ψ), considering the frame of view 1 as the reference, as follows:

$$\begin{aligned} \phi &= \alpha_{12} - \alpha_{21} + \pi \\ \psi &= \arctan(t_2/t_1) = \alpha_{12}. \end{aligned} \tag{3}$$

B. Use of 1D homographies within the proposed framework

The 1D points we use to obtain the homographies and the planar motion arise from taking the angular coordinate of the points in an omnidirectional image. Since, by definition, these 1D points are calibrated, the homographies we compute from them will be calibrated and the reconstruction obtained (3) will be directly Euclidean (as opposed to projective).

Our method has requirements regarding the structure of the scene, because we need landmarks situated in two lines in the 2D projective space. In practice, these landmarks come from the orthogonal projections of the 3D landmarks on the motion plane. Thus, lines in this plane will arise from vertical planes in the 3D scene (Fig. 1). Man-made environments commonly contain vertical planes, which facilitates the applicability of our method. Still, even when there are no real vertical planes in the environment, the technique we propose can be functional. When a considerable number of points are extracted and matched between views, as is typically the case in practice, 1D homographies can be found, emerging from 3D points belonging to virtual planes [16], [17].

As mentioned above, using 1D homographies we obtain the relative angles between two images, which can alternatively be expressed by a rotation and a translation in 2D. Given that this is a purely image-based reconstruction, the translation is obtained up to an unknown scale factor. However, our multirobot control scheme requires the knowledge of the distance between robots (i.e. the scale of the translation). The following section addresses this issue through the use of a state observer.

III. State observer

We discuss in this section how the distance between two robots is estimated in our method. As stated above, this parameter is a degree of freedom in the computed planar

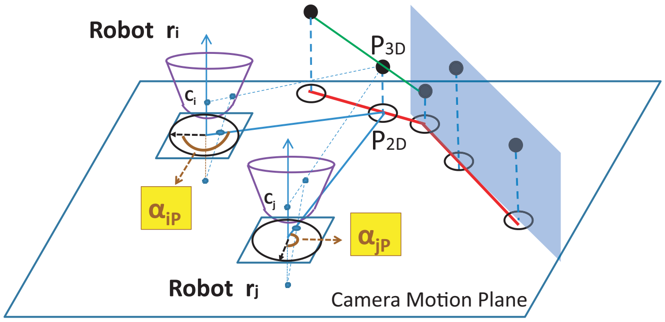

Fig. 1. Origin of the 1D points used to compute the 1D homographies. Two omnidirectional cameras, C_i and C_j, carried by robots r_i and r_j, are shown. The angular components, α_{iP} and α_{jP}, of the camera projections of the scene point P_{3D} generate the 1D projections of the equivalent point P_{2D} belonging to the camera motion plane. The resulting 1D points in projective coordinates for each camera are $(\sin\alpha_{iP}, \cos\alpha_{iP})^T$ and $(\sin\alpha_{jP}, \cos\alpha_{jP})^T$, respectively. Lines in the motion plane arise from vertical planes or non-vertical lines, real or virtual, in the $3D$ space.

motion reconstruction, which results from the estimated angles of the epipoles between robots. Still, it is possible to measure the distance by using the knowledge of the motion performed by the robots, together with these epipole angles. Then, in order to obtain a better estimation of the distance parameter, we integrate its measurements in a state observer, which is described in the following.

Suppose that two robots (r_i and r_j) with unicycle kinematics and commanded with linear and angular velocities $(v_i, \omega_i), (v_j, \omega_j)$ are neighbors, i.e. there is a communication link between them, and that it is possible to compute the epipoles between them using the 1D homography-based method described in the previous section. Then, the dynamics of the angle of the epipole α_{ij} and of the distance between them (ρ_{ij}) are as follows (see Fig. 2) [2]:

$$\dot{\rho}_{ij} = -v_i \cos\alpha_{ij} - v_j \cos\alpha_{ji} \tag{4}$$

$$\dot{\alpha}_{ij} = \frac{v_i \sin\alpha_{ij} + v_j \sin\alpha_{ji}}{\rho_{ij}} - \omega_i. \tag{5}$$

We use the two equations above to design a reduced-state observer for the variable ρ_{ij}. Notice that we can obtain measurements of its value through:

$$\rho_{ij}^m = \left| \frac{v_i \sin\alpha_{ij} + v_j \sin\alpha_{ji}}{\dot{\alpha}_{ij} + \omega_i} \right|. \tag{6}$$

Then, we propose to use a *Luenberger observer* [18] by injecting these measurements in the model of the system:

$$\dot{\hat{\rho}}_{ij} = -v_i \cos\alpha_{ij} - v_j \cos\alpha_{ji} + L_o(\hat{\rho}_{ij} - \rho_{ij}^m), \tag{7}$$

where L_o is the gain of the observer. Thus, $\hat{\rho}_{ij}$ is an improved estimate of the distance between the two robots. The epipoles are known to be variables that can be computed very precisely and robustly. This makes them a good practical choice when considering the variables to use in the design of an observer.

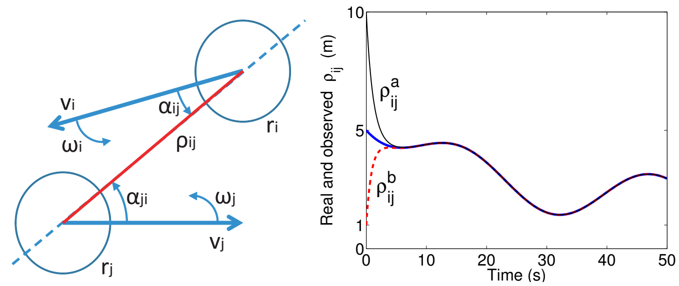

Fig. 2. Left: variables used in the observer of the distance between two robots, r_i and r_j. The angles of the epipoles, α_{ij} and α_{ji}, and the velocities of the two robots, (v_i, ω_i) and (v_j, ω_j), are used to estimate the distance between them, ρ_{ij}. Right: sample simulation of the observer, displaying the evolutions of the real value of ρ_{ij} (thick solid line) and of the value given by the observer of this variable for two cases: an initial overestimation (ρ_{ij}^a) and an initial underestimation (ρ_{ij}^b).

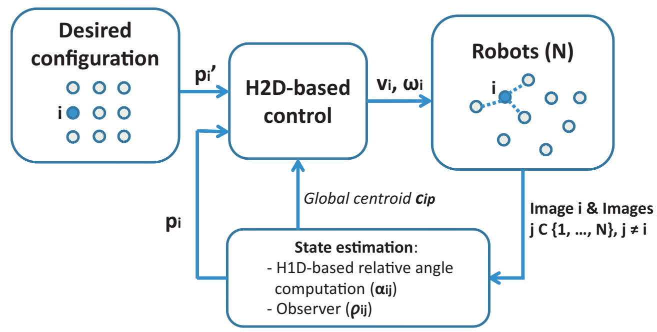

Fig. 3. Overall scheme of the proposed method for multirobot control.

The performance of the employed observer is illustrated in Fig. 2, showing its convergence to the real value in simulation.

There are situations in which the estimation in (6) becomes ill-conditioned or singular. We can deal with these cases, which can be detected, by simply not injecting ρ_{ij}^m into the observer whenever this estimate provides unreliable values. Notice that since the different inter-robot distances are not independent, it would be possible to use distributed estimation algorithms which combine the measurements together in order to obtain a more reliable value for each of them.

IV. Control Strategy

An overview of the full system we propose to carry out the control of the group of robots is depicted in Fig. 3. Next, we describe in particular the way in which the motion control commands for each robot are computed. Notice that using the methods described in the sections above, we can obtain both the relative angles and the distances between neighboring robots. This provides each robot with a metric reconstruction of the positions of other robots.

The approach we propose to control the motion of the robots is inspired by our previous work [19]. In that work, an image-based centralized visual control method was used to bring a set of robots to a formation shape on a planar ground. Using a single camera observing the robot team, a 2D homography was computed from the image of the robots in the desired configuration and the current image. This homography mapped the positions of the robots in the current image to a new set of positions which were used as the goals for the image-based control. Thanks to the particular way in which the homography was defined, the technique minimized at all times the sum of squared distances the robots had to travel for the set to reach the desired configuration. Moreover, we showed in that work that multiple cameras implementing the proposed approach could be used to carry out the desired task in a partially distributed control scheme.

Here, we use a similar idea, i.e. we address the control task through the computation of an appropriate geometric transformation between the sets of current and reference positions. Unlike in the approach of [19], now these positions will not be calibrated image points, but rather the metric 2D positions of other robots expressed in the reference frame of the camera carried by each robot.

We assume that robot r_i, $i = 1, ..., N$, knows at each time these relative positions for a given set of robots I_i, $card(I_i) \leq N-1$. The control interactions between robots can then be described by an undirected formation graph, G_f, wherein every robot r_i is linked to all the robots in the set I_i. Let us denote as $\mathbf{p'}$ and $\mathbf{p}$ the sets of desired positions and current positions of the N robots, respectively, expressed in two arbitrary Cartesian reference frames. Then, robot r_i knows a set of positions in the reference configuration (which we call $\mathbf{p'_i} = \{\mathbf{p'_{ii}}, \mathbf{p'_{ij}}, ..., \mathbf{p'_{ik}}\}$) and a set of positions in the current configuration of the group of robots (which we call $\mathbf{p_i}(t) = \{\mathbf{p_{ii}}(t), \mathbf{p_{ij}}(t), ..., \mathbf{p_{ik}}(t)\}$). Since these positions are expressed in the reference frame of r_i, it is clear that $\mathbf{p'_{ii}} = (0,0)^T$ and $\mathbf{p_{ii}}(t) = (0,0)^T$. The control strategy followed by each robot is based on using its known sets of positions to compute a 2D transformation having a particular parametrization. A general Euclidean 2D transformation, $\mathbf{H_e} \in \mathbb{R}^{3\times3}$, relates two sets of points through a rotation (ϕ_e) and translation ($\mathbf{t} = (t_{xe}, t_{ye})^T$:

$$\mathbf{H_e} = \begin{bmatrix} \cos\phi_e & \sin\phi_e & t_{xe} \\ -\sin\phi_e & \cos\phi_e & t_{ye} \\ 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

Similarly to the work [19], we wish to compute a 2D Euclidean transformation, but we will derive it using a parametrization simpler than (8). This requires a prior translation of the positions in the two sets. Here, this translation is such that the points are expressed with respect to the *global* centroids. These centroids, $\mathbf{c_{ip'}}$ for $\mathbf{p'}$ and $\mathbf{c_{ip}}$ for $\mathbf{p}$, are also expressed in the reference frame of r_i. Thus, we have:

$$\mathbf{p'_{ic}} = \mathbf{p'_i} - \mathbf{c_{ip'}}\ , \qquad \mathbf{p_{ic}} = \mathbf{p_i} - \mathbf{c_{ip}}\ . \tag{9}$$

Then, the sets of positions $\mathbf{p'_{ic}}$ and $\mathbf{p_{ic}}$ are used to compute a transformation constrained to having the following form:

$$\mathbf{H_{il}} = \begin{bmatrix} h^l_{11} & h^l_{12} & 0 \\ -h^l_{12} & h^l_{11} & 0 \\ 0 & 0 & h^l_{33} \end{bmatrix} \sim \begin{bmatrix} s\cos\phi_l & s\sin\phi_l & 0 \\ -s\sin\phi_l & s\cos\phi_l & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Unlike (8), the transformation (10) is not Euclidean. However, as shown in [19], it can be computed linearly using SVD and allows to obtain the least-squares Euclidean transformation that we are looking for, $\mathbf{H^d_i}$, as follows:

$$\mathbf{H^d_i} = \mathbf{H_{il}} \cdot diag(1/s, 1/s, 0). \tag{11}$$

Then, robot r_i uses this computed transformation to obtain its desired position on the plane:

$$\mathbf{p^d_i} = \mathbf{H^d_i}\mathbf{p'_{ic}} + \mathbf{c_{ip}}. \tag{12}$$

In order to drive r_i to the desired location, denoting $\mathbf{p^d_i} = (p^d_{ix}, p^d_{iy})^T$, we use the following position-based control law:

$$v_i = k_v \cdot ||\mathbf{p^d_i}|| \tag{13}$$

$$\omega_i = k_\omega \cdot atan2(p^d_{iy}, p^d_{ix}), \tag{14}$$

where k_v and k_ω are appropriately adjusted control gains.

As shown in [19], if a centralized system is assumed (or, equivalently, if every robot knows the relative positions of all the other robots), this strategy defines desired positions that satisfy the team's desired configuration while minimizing the sum of their squared distances from the robots' current positions. However, a key observation is that the method can also be implemented in a decentralized fashion, with each robot interacting only with a subset of the team.

The control we propose results in each robot always moving towards a position situated at the desired distance from that robot to the global centroid, while getting closer to the partial desired configuration (i.e. the one including only its neighbors in the formation graph). By continuously looking to place the robots at their desired distances to the centroid, the desired overall cohesiveness of the robotic group is always sought and maintained, even when the specific target shape has not yet been reached. This behavior exhibited by the robots is encapsulated in a simple and efficient manner by the 2D transformation we employ. We have observed that it is required to use the global centroid so as to ensure convergence of our control. Otherwise, instability may occur depending on the formation graph and the geometry of the desired configuration. Thus, the only global information that every robot needs to have is an estimation of the centroid of the whole group. This is a usual requirement for coordination in multirobot systems. The centroid information is not costly to communicate among robots and can be computed in a distributed fashion, as shown in several recent works [20], [21]. Our approach can make use of these available methods. We assume that the centroid will vary slowly enough that any dynamic effects in its estimation process can be ignored, which is reasonable in practice.

Once the centroid is known, the control of robot r_i relies only on the relative positions of the robots in I_i. Thus, the multirobot control system we propose is fully distributed. Furthermore, we only require the formation graph G_f to be connected. This modest connectivity requirement is an advantage from the standpoint of system scalability. Observe that in practice, two robots will interact directly if they can communicate and compute their relative motion from common

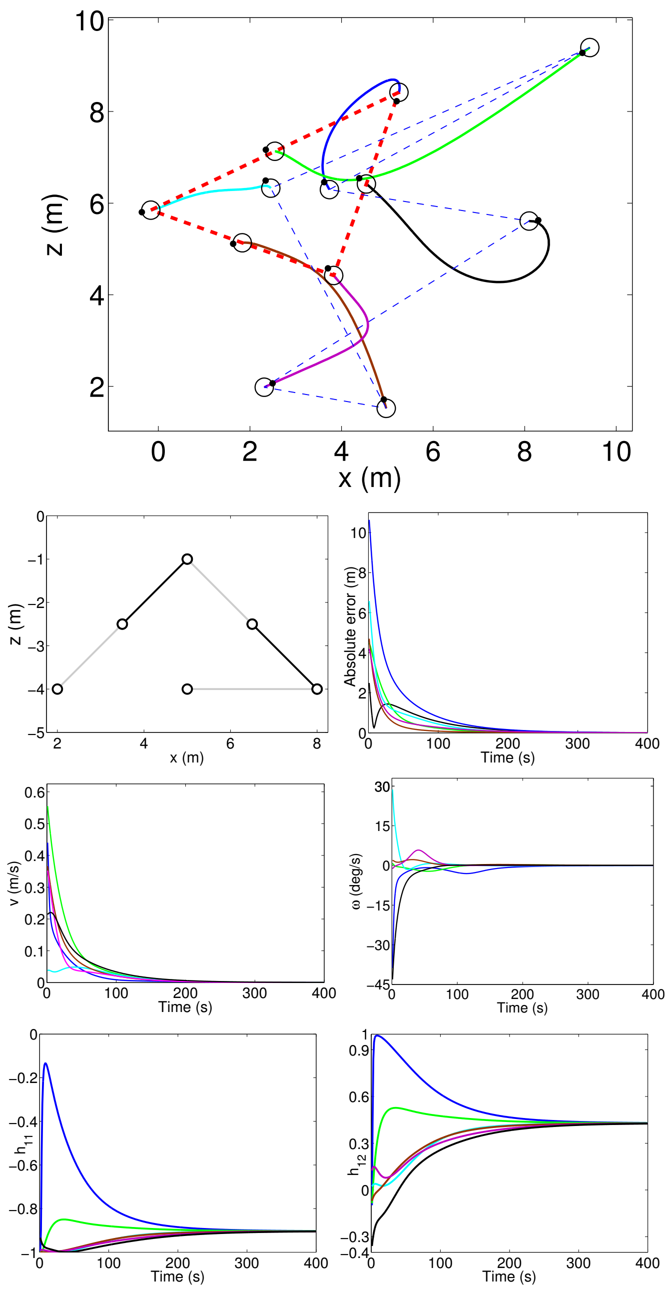

Fig. 4. (Best viewed in color) Simulation results for the proposed control with a six-robot set having to reach a triangle-shaped configuration. Top: Robot paths. The initial positions are joined by thin dashed lines, while the final ones are joined by thicker dashed lines. Second row: reference configuration, with lines showing the formation graph links between robots (left). Evolution of the distances from the current robot positions to the ones given by the desired rigid transformation computed using all the robots (right). Third row: evolutions of the robots' linear (left) and angular (right) velocities. Bottom row: evolution of the two elements, h_{11} (left) and h_{12} (right), of the rigid transformation $\mathbf{H_i^d}$ computed by each robot i.

visual information. These direct robot interactions will give rise to an underlying graph, possibly dynamic, which may not coincide with the static graph G_f. In any case, we require that graph also to be connected for all time.

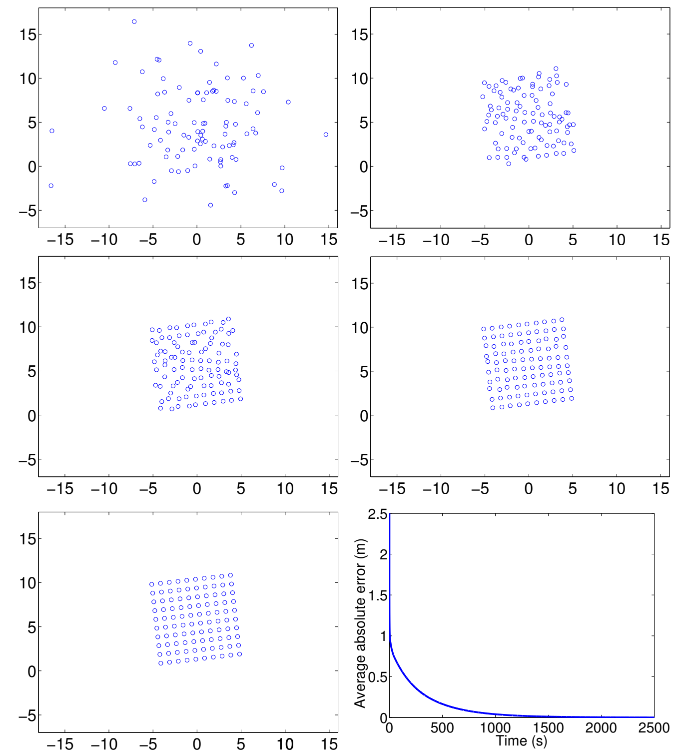

Fig. 5. Results from a simulation of the proposed control strategy with one hundred robots having to reach a square grid-shaped configuration. Top-left: initial positions of the robots. Top-right: robot positions after 100 iterations. Middle row: robot positions after 400 iterations (left) and 1000 iterations (right). Bottom-left: final robot positions, after 2500 iterations. Bottom-right: evolution of the average distances from the current robot positions to the ones given by the desired rigid transformation computed using all the robots.

V. Simulations

In this section, we present simulation results to evaluate the performance of the proposed approach. We assume that each robot r_i is able to obtain the relative position of the robots in I_i, by using the 1D homography-based method described in section II together with the observer defined in section III. We also consider that the position of the centroid of the whole group is available to every robot. For the first simulation example we present, we consider a team of six robots having to reach a desired triangle-shaped configuration from arbitrary initial positions. The formation graph G_f is chosen to be the sparsest possible, i.e. a chain of links connecting all the nodes.

The results are illustrated in Fig. 4. We represent the paths followed by the robots when our control strategy is employed, and the robot velocities. As can be seen from these plots, the robots exhibit a good behavior in terms of the smoothness of their trajectories, and the distributed control scheme brings the team to the target configuration. The errors between the robots' current positions and the ones given by the rigid mapping computed using all the robots (i.e. the mapping obtained with global information) are also depicted. Our control, even if based on partial information, provides satisfactory error regulation performance. The figure displays as well the evolution of the two components of the rigid 2D transformation computed by each robot to obtain its control commands. Since all of these transformations are computed with respect to the common global centroid, it is clear that they must converge to a unique, common transformation when the robot set reaches the target configuration.

The second example we illustrate features a large robotic team (one hundred agents). In this case, the robots are required to attain a square grid shape. We consider that the connected formation graph is such that each robot is linked to only two other robots, selected among its nearest neighbors in the desired configuration. From the results shown in Fig. 5, it can be observed that the desired cohesiveness of the set is rapidly achieved (see the 100 iterations plot) by our approach. The robots finally reach the target configuration. Convergence is slowed down in this case due to the large size of the group and the sparse formation graph considered. The bottom-right plot in the same figure illustrates this behavior. Throughout extensive simulations, the control was consistently observed to converge with varying numbers of robots, geometric configurations and formation graphs.

VI. Discussion and Future Work

We have presented a distributed visual control method for a team of robots moving on the ground plane. The proposed approach relies on the computation of 1D homographies from the visual information shared by the robots. A position-based control strategy implemented by each robot using the computed relative positions of its neighbors makes the group reach a desired configuration.

Works addressing the formation control problem for mobile robots often require the knowledge of the robots' full state [3], [5], or at least their orientation [4], expressed in a common reference frame. In contrast, our method is coordinate-free, i.e. each robot operates referring only to its own coordinate frame. This makes the approach flexible and more feasible in the case where the information must come only from vision sensors carried by the robots. Also, we do not consider a leader robot with respect to which the formation is defined, unlike [1], [6], [9].

Regarding vision-based related works, the control methods [8]–[10], [19] are all image-based, whereas our presented approach is position-based. While this requires us to estimate the relative states of the robots, it will lead, especially in the cases where the visual sensing is distributed, to higher accuracy. Since our control method is completely distributed, it will be more scalable and robust than centralized [8] or mixed [19] approaches. The works [9], [10] propose distributed methods that do not use any sort of inter-agent communication, but require each robot to visually sense and identify their neighbors. Our use of multiple-view models avoids these tasks, which can be challenging to carry out and provide limited accuracy and flexibility.

Several issues relevant to this work could be interesting to explore in the future: for instance, how to maintain the required connectedness of the communications graph, the problem of dynamically assigning the robots to the positions in the desired configuration to improve the efficiency, or the incorporation of a collision avoidance strategy.

References

[1] J. Desai, J. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 6, pp. 905–908, 2001.

[2] J. Marshall, M. Broucke, and B. Francis, "Formations of vehicles in cyclic pursuit," *IEEE Transactions on Automatic Control*, vol. 49, no. 11, pp. 1963–1974, 2004.

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

[4] 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.

[5] N. Michael and V. Kumar, "Planning and control of ensembles of robots with non-holonomic constraints," *International Journal of Robotics Research*, vol. 28, no. 8, pp. 962–975, 2009.

[6] 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.

[7] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley, "Multi-robot system for artistic pattern formation," in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 4512–4517.

[8] G. López-Nicolás, M. Aranda, Y. Mezouar, and C. Sagüés, "Visual control for multirobot organized rendezvous," *IEEE Trans. Sys., Man, and Cybern., Part B: Cybern.*, vol. 42, no. 4, pp. 1155 –1168, 2012.

[9] 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.

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

[11] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[12] G. Zhang, H. Zhang, and K.-Y. K. Wong, "1D camera geometry and its application to circular motion estimation," in *Proceedings of the British Machine Vision Conference, vol. I*, 2006, pp. 67–76.

[13] M. Aranda, G. López-Nicolás, and C. Sagüés, "Planar motion estimation from 1D homographies," in *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2012, pp. 329–334.

[14] K. Åström and M. Oskarsson, "Solutions and ambiguities of the structure and motion problem for 1D retinal vision," *Journal of Mathematical Imaging and Vision*, vol. 12, pp. 121–135, 1999.

[15] O. Faugeras, L. Quan, and P. Sturm, "Self-calibration of a 1D projective camera and its application to the self-calibration of a 2D projective camera," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 10, pp. 1179–1185, 2000.

[16] E. Malis and F. Chaumette, "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *Int. Journ. Comp. Vis.*, vol. 37, no. 1, pp. 79–97, 2000.

[17] G. López-Nicolás, J. Guerrero, and C. Sagüés, "Multiple homographies with omnidirectional vision for robot homing," *Robotics and Autonomous Systems*, vol. 58, no. 6, pp. 773–783, 2010.

[18] D. Luenberger, "Observing the state of a linear system," *Military Electronics, IEEE Transactions on*, vol. 8, no. 2, pp. 74–80, 1964.

[19] M. Aranda, Y. Mezouar, G. López-Nicolás, and C. Sagüés, "Partially distributed multirobot control with multiple cameras," in *American Control Conference*, 2013, pp. 6323–6329.

[20] R. Aragüés, L. Carlone, C. Sagüés, and G. Calafiore, "Distributed centroid estimation from noisy relative measurements," *Systems & Control Letters*, vol. 61, no. 7, pp. 773–779, 2012.

[21] M. Franceschelli and A. Gasparri, "Decentralized centroid estimation for multi-agent systems in absence of a common reference frame: A convergence analysis," in *IEEE Conference on Decision and Control*, 2012, pp. 2071–2076.