IROS Workshop on Visual Control of Mobile Robots (ViCoMoR 2012)
October 11th, 2012, Vilamoura, Algarve, Portugal

Distributed Policies for Neighbor Selection in Multi-Robot Visual Consensus

Eduardo Montijano Johan Thunberg Xiaoming Hu Carlos Sagues

Abstract— **In this paper we propose a distributed algorithm for choosing the appropriate neighbors to compute the control inputs of a team of robots. We consider a scheme where the motion of each robot is decided using nearest neighbor rules. In this scheme each robot is equipped with a camera and can only exchange visual information with a subset of the robots. Using the information provided by their neighbors, the robots compute their control inputs, eventually reaching a consensus in their motion. However, if a robot has too many neighbors (e.g., a star topology), then it will require a long time to process all the received information, leading to long loop times or synchronization problems. In the paper we provide two distributed policies for the robots to select at each iteration the information of a fixed number of neighbors. In both cases we demonstrate convergence to the consensus with a considerable reduction on the amount of required computations. Simulations in a virtual environment show the effectiveness of the proposed policies.**

I. INTRODUCTION

The idea of multiple robots working in cooperation to achieve a common goal is of high interest in many tasks, such as exploration, surveillance or transportation. Multi-robot systems can perform these tasks with more robustness or in less time than one robot working alone. On the other hand, in order to carry on with these tasks, the robots need to be able to move in coordination.

A generalized problem in this context is the problem of reaching a consensus by all the robots. From the control perspective, the consensus problem [14] consists of making a team of robots to move all together in a common direction, with the peculiarity that this goal is achieved by each robot using only partial information given by the nearest-neighbors in the team. In this way all the robots play the same role in the formation, conferring the system a natural robustness against changes in the topology and individual failures. A key aspect left aside in most of the existing work in this topic, e.g., [2], [6], [12], [15], is how the robots estimate their neighbors positions to control their motion.

Vision sensors can play a fundamental role in this part of the process due to the big amount of information that images contain. Additionally, all the research done in the field of computer vision during the past decades can be exploited in a multi-robot framework in order to achieve the desired goal. Some works consider a single camera and a central unit to control all the robots [7]. Distributed solutions using omnidirectional cameras can be found in [13], [16] where the robots can see all their neighbors. If the robots are equipped with monocular cameras with limited field of view, then the observation of all the neighbors may not always be possible. A leader-follower solution is adopted in [5], where each robot only needs to observe another robot, leading to tree configurations. Geometry constraints are used in [11] to allow the network to be configured in any arbitrary topology.

In the latter approach each robot computes its control input using the epipoles between its current image and the images of its neighbors in the communication graph. The use of the epipolar constraint presents some advantages over other approaches. First of all, it has been successfully used to control the motion of a single robot on several occasions [1], [8], which gives this constraint reliability to be used in a multi-robot context. Secondly, the robots can reach the consensus even if they are not directly observing each other as long as they have common observations of the environment. Lastly, the controller does not impose any constraint on the network topology as each robot can compute as many pairs of epipoles as neighbors in the communication graph it has.

On the other hand, in the approach presented in [11], the number of neighbors determines the amount of time each robot will require to compute its control input. In a distributed scenario, if one robot has many neighbors, e.g., a star topology, with one robot connected to all the others, then it will receive many images. Processing all the images may take a long time, depending on the computation capabilities of the robots. This can lead to long times in the control loop or even to synchronization problems between the robots with different number of neighbors. Therefore, additional mechanisms are required to keep the amount of computations under control for all the robots.

In this paper we contribute to the state of the art presenting two distributed policies that allow the robots to select only a subset of their neighbors to compute the control input. In this way the team is still able to reach the consensus but the computational demands of each robot are bounded and equal. Additionally, we discuss the convergence of the considered controller for directed graphs.

The rest of the paper is organized as follows: In section II we review the distributed control law using epipoles to reach the consensus of the team of robots. All the formal details about the distributed policies for neighbor selection are explained in section III. Section IV shows simulations in a virtual environment where the two proposed policies

E. Montijano is with Centro Universitario de la Defensa (CUD) and Instituto de Investigación en Ingeniería de Aragón (I3A), Zaragoza, Spain.
J. Thunberg and X. Hu are with Department of Math, Division of Optimization and Systems Theory, Royal Institute of Technology (KTH), Sweden.
C. Sagues is with Departamento de Informática e Ingeniería de Sistemas - Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain.

are tested and compared with the standard distributed controller. Finally, in section V the conclusions of the work are presented.

II. Distributed Consensus using Epipoles

In this section we review the distributed controller based on the epipolar geometry to achieve the consensus. For additional details we refer the reader to [11].

We consider a set $\mathcal{V}$ of N homogeneous autonomous robots. Communications between the robots are defined with a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{E}$ the set of communication links. In this way, if robots i and j are able to exchange messages with each other, then $(i, j) \in \mathcal{E}$. The neighbors of robot i are defined as $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$.

The robots move on the plane with non-holonomic motion constraints. Given two robots, i and j, their relative positions can be defined by a distance, r_{ij}, a bearing angle, ψ_{ij}, and relative orientation, θ_{ij}. The goal of the consensus problem is to make all the robots achieve the same orientation, i.e., $\theta_{ij} \to 0, \forall i, j \in \mathcal{V}$, as $t \to \infty$. To achieve this goal, each robot has two control inputs, v_i and w_i, which are the linear and angular velocity respectively. Since the linear velocity is not required to make the robots achieve the consensus, along the paper we consider it constant for all the robots, $v_i = v \geq 0, \ \forall i$.

In our setup all the robots are equipped with pinhole monocular cameras with limited field of view. We assume that all the robots have identical cameras onboard, with unknown calibration matrix equal to $\mathbf{K} = \text{diag}(\alpha, \alpha, 1)$, with $\alpha > 0$, the focal length of the camera. For any pair of robots, i and j, the lack of calibration implies that r_{ij}, ψ_{ij} and θ_{ij} are not directly available. The output of the system is instead defined by the epipoles of the images acquired by them (see Fig. 1). The robots exchange their images and use the epipolar constraint [10] to compute e_{ij} and e_{ji}, the epipoles in the two images. Specifically, due to the planar motion, we are only interested in the x-coordinate of the epipoles, which satisfies

$$e_{ijx} = \alpha \tan(\psi_{ij}), \ e_{jix} = \alpha \tan(\psi_{ij} - \theta_{ij}). \tag{1}$$

For simplicity purposes, in the following we use e_{ij} and e_{ji} to refer to the expressions in equation (1).

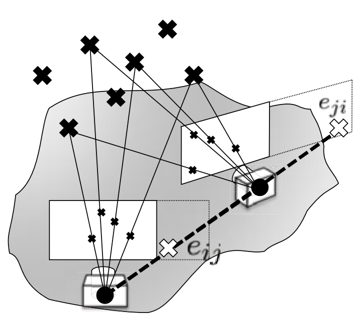

Fig. 1: The robots use the epipoles between their images to compute the control input.

Given a pair of neighbor robots, by eq. (1), a necessary condition for the consensus is that their epipoles must be equal, $\theta_{ij} = 0 \Rightarrow e_{ij} = e_{ji}$. To reach this objective the control input w_i of each robot is defined as:

$$w_i = K \sum_{j \in \mathcal{N}_i} w_{ij}, \tag{2}$$

where $K > 0$ is the controller gain and w_{ij} is the misalignment in the epipoles, defined as

$$w_{ij} = \begin{cases} d_{ij} & \text{if } |d_{ij}| \leq \frac{\pi}{2} \\ -\text{sign}(d_{ij})(\pi - |d_{ij}|) & \text{otherwise} \end{cases}, \tag{3}$$

with

$$d_{ij} = \arctan(\frac{e_{ij}}{\beta}) - \arctan(\frac{e_{ji}}{\beta}) \in (-\pi, \pi], \tag{4}$$

and $0 < \beta < \infty$ some fixed positive constant to choose. Note that, if $\beta = \alpha$, then the setup is calibrated, $d_{ij} = \theta_{ij}$, and the relative orientation between the robots can be computed from the epipoles. However, we assume that this is not the case and $\beta \neq \alpha$.

The following result determines the conditions required for the controller to converge to the consensus:

***Theorem** 2.1 (Theorem 3.2 [11]):* Let the robots be initially oriented in such a way that $|\theta_{ij}| \leq \theta_M < \pi/2, \ \forall i, j \in \mathcal{V}$. If the robots use the control law (2) with β satisfying

$$\alpha \tan(\frac{\theta_M}{2}) < \beta < \frac{\alpha}{\tan(\frac{\theta_M}{2})}, \tag{5}$$

then $\lim_{t \to \infty} \theta_{ij} = 0, \forall i, j \in \mathcal{V}$, i.e., the system will reach consensus. ■

Additionally, the system is robust to changes in the communication topology, as long as the following assumptions are satisfied.

***Assumption** 2.1:* There exists a lower bound, $\delta > 0$, on the time between two consecutive changes in the topology. Denoting $t_k, \ k \in \mathbb{N}$, the discrete time instants when the topology changes, then $t_{k+1} - t_k \geq \delta, \ \forall k$.

***Assumption** 2.2:* There exists a positive time period T such that, for any instant of time, t, the collection of communication topologies in the time interval $(t, t + T)$ is jointly connected.

The problem with the aforementioned controller is the amount of computations that the robots require in order to compute the epipoles between them and all their neighbors. If one robot has too many neighbors then it will have to compute many epipoles. For that reason additional mechanisms are required to keep the computational demands of all the robots bounded and similar.

Along the rest of the paper we will assume that the communication graph is fixed and the conditions in Theorem 2.1 regarding β and the initial orientations are satisfied. Since the neighbor policies will select different neighbors at each iteration, the assumptions regarding the changes in the communication topology will be required to prove the convergence.

III. Distributed Policies for Neighbor Selection

In this section we propose two distributed policies to select, from the subset of neighbors, which one each robot should choose to compute the epipoles. The first policy chooses at each iteration the robot that was not selected for the longest time. The second policy chooses at each iteration the neighbor that supposedly has the orientation farthest away. For both policies we prove convergence to the consensus.

Before explaining in detail our policies for neighbor selection, let us note that by selecting a subset of the neighbors what we are doing in practice is just changing the communication topology at each iteration. However, the proposed policies do not ensure a bi-directional selection, that is robot the fact that robot i chooses j to compute the epipoles does not imply that robot j chooses robot i as well. As a consequence, the communication topologies need to be modeled with time-varying directed graphs. Nevertheless, the proposed controller will also reach the consensus if Assumptions 2.1 and 2.2, see, e.g., [4]. Therefore, it will be enough to prove that the proposed policies ensure that Assumptions 2.1 and 2.2 are satisfied to reach the consensus.

In the following we present the two policies and formally prove the convergence to the consensus.

A. Policy 1: Choose the neighbor that was not selected for the longest time

The first policy we propose consists of selecting a different neighbor at each iteration. Specifically the one that was not selected for the longest time. Let each robot handle a vector $\mathbf{N}_i(t) = [N_{i1}(t), \ldots, N_{iN}(t)]$, with $N_{ij}(t)$ representing the number of communication rounds that has passed since the last time that robot i chose robot j as the selected neighbor to compute the epipoles.

Initially, $N_{ij}(0) = 0$ for all j. Then, at time t, the neighbor selected by robot i, denoted by $j(t)$, will be

$$j(t) = \arg_{j \in \mathcal{N}_i} \max N_{ij}(t), \tag{6}$$

and the control input of the robot

$$w_i(t) = K w_{ij(t)}. \tag{7}$$

Once the robot has computed the epipoles and the control input, it updates the vector $\mathbf{N}_i(t)$ with the following rule

$$N_{ij}(t+1) = \begin{cases} 0 & \text{if } j = \{j(t), i\} \\ N_{ij}(t) + 1 & \text{otherwise} \end{cases}, \tag{8}$$

so that it ensures that $j(t)$ will not be chosen again until all the other possible neighbors have been chosen once.

Proposition *3.1:* Let all the robots select, at each iteration, one neighbor to compute the epipoles using equations (6), (8) and move using the controller in eq. (7) considering only this neighbor. Then, the system will reach the consensus.

Proof: Since we are assuming that the communication topology is fixed, $\mathcal{G}$ necessarily is connected and the neighbors of each robot remain constant the whole time. At any iteration, t, each robot selects only one of its neighbors, which may lead to a disconnected digraph, $\mathcal{G}(t) \subseteq \mathcal{G}$. However, in the following iterations the selected neighbor of all those robots with more than one will be different. Denoting $\mathcal{N}_{\max} = \max |\mathcal{N}_i|$, we can see that for any t

$$\mathcal{G}(t) \cup \mathcal{G}(t+1) \cup \ldots \cup \mathcal{G}(t + \mathcal{N}_{\max}) = \mathcal{G},$$

which means that Assumption 2.2 is satisfied when using this policy. Considering that the time required to compute the epipoles is not zero, Assumption 2.1 is also satisfied and then we can conclude that the time varying evolution of the graph satisfies the conditions to reach the consensus. ■

B. Policy 2: Choose the neighbor with more misalignment

The second policy we propose is designed to reduce the orientation error with the neighbor that supposedly is further away at each iteration.

Let each robot have a vector $\hat{\mathbf{d}}_i(t) = [\hat{d}_{i1}(t), \ldots, \hat{d}_{iN}(t)]$, with $\hat{d}_{ij}(t)$ being the last value of d_{ij} computed by robot i using the information provided by robot j, equation (4). Initially, $\hat{d}_{ij}(0) = \infty$ for all j. The neighbor selected at each iteration is chosen by

$$j(t) = \arg_{j \in \mathcal{N}_i(t)} \max \hat{d}_{ij}(t), \tag{9}$$

that is, the one with the estimated most misalignment at current time. The control input of each robot is then assigned as in eq. (7).

With the epipoles computed, the update of $\hat{\mathbf{d}}_i(t)$ executed in this case is

$$\hat{d}_{ij}(t+1) = \begin{cases} d_{ij} & \text{if } j = j(t) \\ \hat{d}_{ij}(t) & \text{otherwise} \end{cases}, \tag{10}$$

with d_{ij} the value computed in equation (4).

Proposition *3.2:* Let all the robots select, at each iteration, one neighbor to compute the epipoles using equations (9), (10) and move using the controller in eq. (7) considering only this neighbor. Then, the system will reach the consensus.

Proof: With this policy we cannot ensure that the robots are changing the selected neighbors at each iteration. Let us assume that none of the robots change the selected neighbor, this means that the topology of the network remains fixed for some time. In such case we already now that the team approaches to the consensus, meaning that $d_{ij} \to 0$ for all i and j that are neighbors in the subgraph defined by the policy, and so $\hat{d}_{ij} \to 0$ for the same set. This means that at some point there will be some robot i and some $k \in \mathcal{N}_i$ such that $\hat{d}_{ij} > \hat{d}_{ik}$ and the robot will change the selected neighbor. Noting that any change of the selected neighbor does not change the rest of values, $\hat{d}_{ij}$, in the network, we can use this argument iteratively to see that all the neighbors of each robot will be selected at some point. After that, using the same arguments as in Proposition 3.1 we conclude that the system will reach the consensus. ■

C. Discussion

There are several advantages of using any of the two proposed policies instead of computing the epipoles with the images sent by all the robots. First of all, the consensus is achieved requiring less computations at each communication round. Each pair of computed epipoles requires an initial step to match the features of the two images plus a robust method to estimate the epipolar constraint, e.g., DLT+RANSAC [10]. By selecting only one neighbor, we are executing this step only once at each iteration instead of $|\mathcal{N}_i|$ times. This computational reduction can be of high interest in situations where the energy of the robots is limited.

Synchronization issues are also solved. Note that the controller requires the images of all the robots to be acquired (approximately) at the same instant. If each robot does not acquire a new image until it has processed all the received information, and assuming that the number of neighbors is not going to be the same for all the robots, then without additional mechanisms to synchronize the network there will appear time discrepancies among the matched images. On the other hand, making all the robots to select only one neighbor to compute the epipoles, will imply similar computation times for all of them, leading to a natural synchronization.

In case the robots want to select more than one neighbor, let us say k neighbors at each iteration, both policies are still be applicable. In the first policy, each robot selects the k neighbors that were not selected for the longest time whereas the in the second policy, each robot selects the k neighbors with the largest value of $\hat{d}_{ij}(t)$. The larger k, the most similar would be the results to the standard case, at the price of more and more computational demands.

IV. Simulations

The properties of the proposed controller are shown in simulations. The experiments have been carried out using Matlab. We have considered a fixed robotic network composed by ten robots with initial positions and orientations depicted in Fig. 2 and communications defined by the black lines. As we can see, there are some robots that have up to five neighbors in the communication graph whereas others only have one or two neighbors. The vision system has been

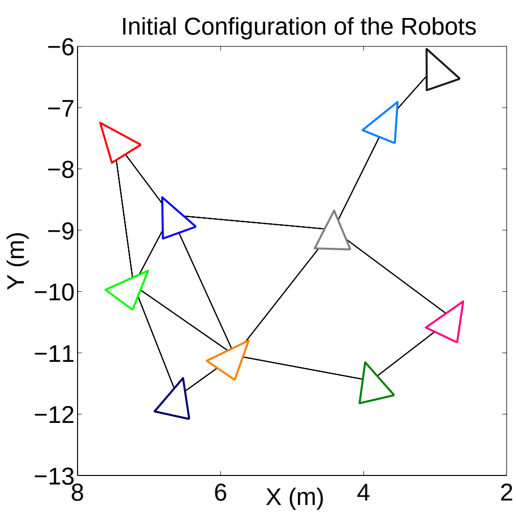

Fig. 2: Initial configuration of the team of robots in the experiments. Black lines represent direct communication between robots.

simulated using the virtual reality toolbox of MatLab. In this way, the robots acquire virtual images of resolution 640×480 pixels depending on their position and orientation. We have extracted SIFT [9] features from the virtual images and the 8 point algorithm with RANSAC [3] to match them in a robust way and to compute the epipoles between pairs of robots. An example of the images acquired by the robots and the features extracted and matched can be found in Figure 3.

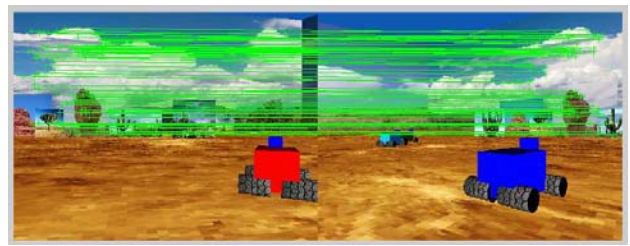

Fig. 3: Example of images acquired by the robots and SIFT matches using the epipolar constraint.

We have executed the controller in eq. (2) without using a neighbor selection policy and the controller (7) using the two policies proposed in section III. The evolution of the orientation of the robots in the three cases can be seen in Figure 4. As expected, all the robots reach the consensus without problems in the three situations. The evolution of the orientation when the information of all the neighbors is used is smoother and the consensus is reached in less time. However, the difference with respect to the other two graphics is negligible and in the simulation we are not considering the real time spent by each robot to compute the epipoles with all the neighbors. The second policy (neighbor with more misalignment) reaches the consensus relatively faster than the first policy, which makes sense because it tries to reduce the error with the robot with most relative orientation.

The control inputs of the robots in each scenario are depicted in Fig. 5. Again, the control inputs when no policies are used are smoother than when using any policy because they are computed using the same set of images the whole time but this simulation is not considering the real time spent to compute the inputs. The second policy also seems to be better than the first one in this aspect.

TABLE I: Computational time (seconds per robot and iteration)

Quantity	No policy	Policy 1	Policy 2
Mean time	12.65	3.80	3.85
Std. dev	5.83	0.77	0.75
Max time	27.99	5.92	5.84
Min time	2.55	1.49	1.28

We have measured the time spent to compute the control inputs when the robots do not use the proposed policies and when they do to point out the real advantages of using a policy to select a subset of neighbors. The computational time spent by each robot at each iteration is depicted in Fig. 6. We can see that using any of the two policies the computational time per iteration and robot remains bounded and similar whereas in the standard case there are big variations in the loop time, depending on the number of

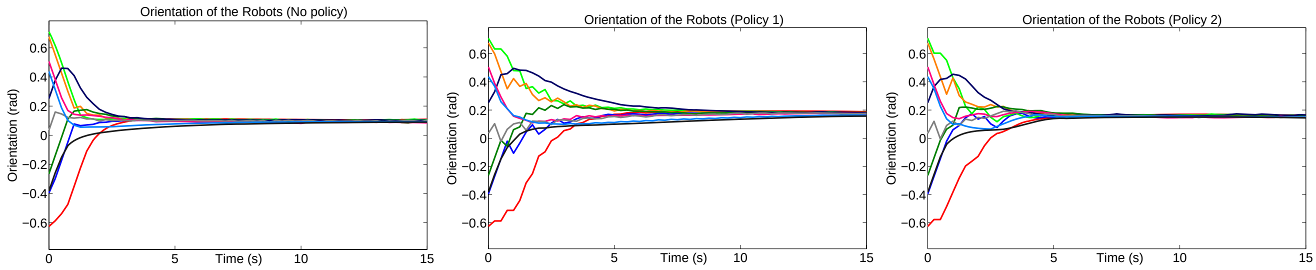

Fig. 4: Orientation of the robots using the distributed controller without any policy to select the neighbors (left), with the policy to choose the neighbor that was not selected for the longest time (middle) and with the policy to choose the neighbor with the most misalignment (right). In the three cases all the robots reach the consensus in a similar time.

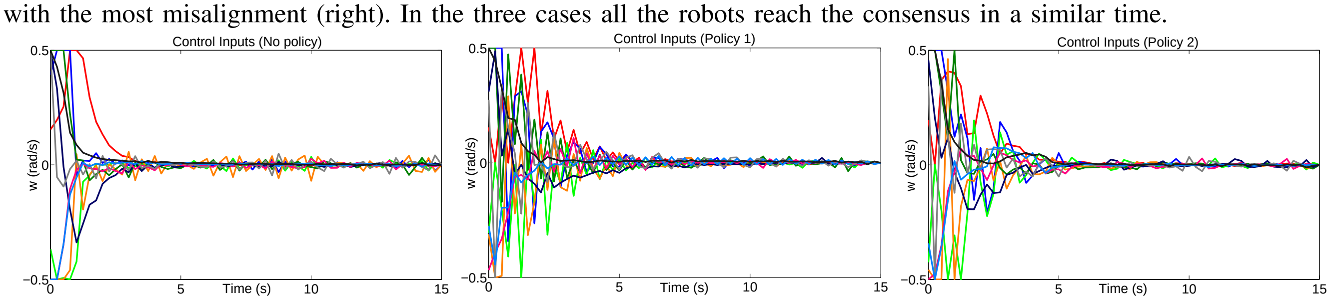

Fig. 5: Control inputs of the ten robots in the three scenarios: without any policy to select the neighbors (left), with the policy to choose the neighbor that was not selected for the longest time (middle) and with the policy to choose the neighbor with the most misalignment (right).

Fig. 6: Computational time spent by each robot at each iteration: without any policy to select the neighbors (left), with the policy to choose the neighbor that was not selected for the longest time (middle) and with the policy to choose the neighbor with the most misalignment (right). As we can see, when the robots use a policy, all of them spend approximately the same computational time, whereas without a policy each robot requires a different time depending on its neighbors.

neighbors of each robot. The statistics of these times are shown in in Table I.

V. Conclusions

In this paper we have proposed two distributed policies for a team of robots to select a subset of their neighbors to compute their control inputs. This selection bounds the time required by the robots to compute the input, making it equal for all of them, avoiding possible synchronization problems that could appear because of the different computation requirements of each robot. The computational reduction is also of high interest when the information shared by the robots is provided by vision sensors, because image processing methods are in general time-demanding. We have proved that both policies ensure convergence to the consensus and we have shown in simulations the benefits of using any of the two approaches compared to the situation in which all the neighbors are considered.

Acknowledgments

This work was supported by the project DPI2009-08126 and partly supported by Swedish Research Council and Swedish Foundation for Strategic Research.

References

[1] R. Basri, E. Rivlin, and I. Shimshoni, *Visual homing: Surfing on the epipoles*, International Journal of Computer Vision **33** (1999), no. 2, 117–137.

[2] J. Cortes, *Global and robust formation-shape stabilization of relative sensing networks*, Automatica **45** (2009), no. 12, 2754 – 2762.

[3] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, Cambridge, 2000.

[4] Q. Hui and W. M. Haddad, *Distributed nonlinear control algorithms for network consensus*, Automatica **44** (2008), no. 1, 2375–2381.

[5] T. Ibuki, T. Hatanaka, M. Fujita, and M. W. Spong, *Visual feedback attitude synchronization in leader-follower type visibility structures*, 49th IEEE Conference on Decision and Control, December 2010, pp. 2486–2491.

[6] A. Jadbabaie, J. Lin, and A. S. Morse, *Coordination of groups of mobile autonomous agents using nearest neighbor rules*, IEEE Transactions on Automatic Control **48** (2003), no. 6, 988–1001.

[7] G. Lopez-Nicolas, M. Aranda, Y. Mezouar, and C. Sagues, *Visual Control for Multi-Robot Organized Rendezvous*, IEEE Transactions on Systems Man and Cybernetics: Part B (2012), to appear.

[8] G. López-Nicolás, C. Sagues, J.J. Guerrero, D. Kragic, and P. Jensfelt, *Switching visual control based on epipoles for mobile robots*, Robotics and Autonomous Systems **56** (2008), no. 7, 592–603.
[9] D. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision **60** (2004), no. 2, 91–110.
[10] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An invitation to 3d vision*, SpringerVerlag, 2004.
[11] E. Montijano, J. Thunberg, X. Hu, and C. Sagues, *Multi-Robot Distributed Visual Consensus using Epipoles*, IEEE Conference on Decision and Control, 2011, pp. 2750–2655.
[12] N. Mostagh and A. Jadbabaie, *Distributed geodesic control laws for flocking of nonholonomic agents*, IEEE Transactions on Automatic Control **52** (2007), no. 4, 681–686.
[13] N. Mostagh, N. Michael, A. Jadbabaie, and K. Daniilidis, *Vision-based, distributed control laws for motion coordination of nonholonomic robots*, IEEE Transactions on Robotics **25** (2009), no. 4, 851–860.
[14] Wei Ren and Randal W. Beard, *Distributed consensus in multi-vehicle cooperative control*, Communications and Control Engineering, Springer-Verlag, London, 2008.
[15] J. Thunberg, E. Montijano, and X. Hu, *Distributed attitude synchronization control*, 50th IEEE Conference on Decision and Control and European Control Conference, December 2011, pp. 1962–1967.
[16] R. Vidal, O. Shakernia, and S. Sastry, *Following the flock [formation control]*, IEEE Robotics and Automation Magazine **11** (2004), no. 4, 14–20.