Hierarchical Congestion Control for Robotic Swarms

Vinicius Graciano Santos and Luiz Chaimowicz

Abstract— **Safe and efficient navigation of robotic swarms is an important research problem. One of the main challenges in this area is to avoid congestion, which usually happens when large groups of robots share the same environment. In this paper, we propose the use of hierarchical abstractions in conjunction with simple traffic control rules based on virtual forces to avoid congestion in swarm navigation. We perform simulated and real experiments in order to study the feasibility and effectiveness of the proposed algorithm. Results show that our approach allows the swarm to navigate without congestions in a smooth and coherent fashion, being suitable for large groups of robots.**

I. INTRODUCTION

The use of large groups of robots has received much attention in recent years. Generally called swarms, these systems employ a large number of simple agents to perform complex tasks such as exploration and mapping of unknown environments, transportation and manipulation of large objects and also distributed sensing and actuation.

Not only limited to robotics, the study of algorithms to control swarms also has a large impact on other fields such as digital games, virtual reality and crowd simulation. Typically, the virtual environments created in these applications are populated by entities that should behave as a coherent group rather than unique individuals. For example, given a virtual model of a building, it is desirable to simulate the behavior of a crowd evacuating the site in case of fire or other emergencies.

A common problem in most of these tasks is swarm navigation. As the dimension of the system's configuration space increases exponentially with the number of agents, the use of traditional path planning algorithms to solve this problem becomes very expensive, even for a small number of robots. Possible solutions include separately planning a path for each agent, considering only its configuration space, or generating global vector fields to control the swarm as a whole. But these solutions generally lead to conflicts in robot trajectories that must be solved. One example is traffic congestion, when a large number of robots moves towards the same region of the environment in the same time interval.

In recent years, new approaches have tried to model the swarm using virtual structures that reduce the dimensionality of the control problem [1]. In general, these structures define formation rules for the agents in order to group them hierarchically, introducing a new level of abstraction to control the swarm since only the virtual structure needs to be explicitly controlled.

This work was partially supported by CNPq and Fapemig.

The authors are with the Vision and Robotics Laboratory (VeRLab), Computer Science Department, Federal University of Minas Gerais, Brazil. emails: {vgs, chaimo}@dcc.ufmg.br

The objective of this paper is to use the benefits of this hierarchical paradigm to develop a control algorithm that minimizes traffic congestions when different groups attempt to pass through a common location in the environment. We realized that most of the work on grouping behaviors and virtual structures tend to concentrate their efforts in defining the structure and the formation rules, in an attempt to minimize the dimension of the resulting configuration space. However, when we start to consider more than one group, we see that it is necessary to explore the combined configuration space formed by them to find an optimal path without congestions, which leads us back to the same original problem. Therefore, traffic control algorithms are still needed due to the increasing configuration space dimension when considering several groups in the same environment.

This paper is organized as follows. Section II discuss some related work in the field. Section III presents the methodology used for grouping the robots and the hierarchical algorithms for the traffic control. Experimental results with both simulated and real robots are presented in Section IV, while Section V brings the conclusion and directions for future work.

II. RELATED WORK

One of the earliest works that considered the problem of controlling a swarm of agents was presented in [2] with the aim of realistically simulating a flock of birds, known as *boids*. In summary, local interactions between agents within a neighboring area define an emergent behavior for the whole swarm. Such interactions can be modeled as a special case of the social artificial potential field method [3], in which a gradient descent technique is used to generate virtual forces that guide the robots towards its goal while locally avoiding obstacles.

A different approach considers the whole group as a single entity, sometimes called virtual structure. The desired motion is commanded to this structure, implicitly controlling the robotic swarm. The works presented in [4] and [5], for example, define controllers that converge and maintain a group of robots in a rigid formation according to a known structure. However, such methods are not scalable to large groups and the geometric relations among robots make formation changes during movement a difficult task.

Deformable structures were presented in [6] and [7] together with artificial potential fields to group and control swarms of robots. In the former, *Probabilistic Roadmaps* [8] were used for planning a navigation path for the structure in an environment with obstacles; whereas in the latter, controllers were designed in order to converge the swarm into a known elliptical region, which was used to escort a vehicle convoy. Instead of considering a single deformable structure, some studies used a set of structures to increase group cohesion and to simplify the path planning problem. In [9], for example, a hierarchical sphere tree was proposed for controlling "crowds of robots". In [10] the path planned for a single agent is extended to a corridor using the clearance along the path, and by changing the characteristics of this corridor it is possible to control a swarm that navigates through it in a desired way.

Based on a mapping of the swarm's configuration space to a lower dimensional manifold, whose dimension is independent of the number of robots, a formal abstraction that allows decoupled control of the pose and shape of a team of ground robots was developed in [1]. This work was extended in [11], to account for three dimensional swarms, and in [12], where a dynamic control model was introduced for similar abstractions. Based on [1], a cooperation mechanism between multiple unmanned aerial and ground vehicles was developed [13], where the UAVs are responsible for estimating the configuration of the ground robots and also for sending control messages to the groups. Merging and splitting behaviors were also studied, as sometimes these maneuvers are necessary for groups to overcome obstacles. Nevertheless, interactions between groups with different goals and congestion control were not addressed. This type of problem can be tackled by applying traffic control rules.

The traffic control problem is an important research topic, being characterized as a *resource conflict* problem [14]. In general, works in this area assume that robots are contained in a structured environment [15], [16], in which they navigate in delimited lanes that meet at intersections, usually where traffic control is performed. This can be accomplished by using a single manager agent [17] or a more robust sensor network [18].

In previous works [19], [20], we proposed distributed algorithms for congestion control in unstructured environments. These algorithms relied on virtual forces and local communication to control the robots. Specifically, the work presented in [20] dealt with the problem of groups moving in opposite directions, which can be regarded as one of the worst case congestion scenarios. Also, recent advances on distributed collision avoidance provided an optimal method that guarantees smooth and collision-free motions under nonholonomic constraints [21].

As mentioned, in this paper we explore the use of a hierarchical paradigm in conjunction with traffic control algorithms. The objective is to take advantage of the ideas and robust solutions that have been developed in both areas to be able to efficiently avoid congestions during swarm navigation.

III. METHODOLOGY

Following the hierarchical architecture proposed in [13], we consider a set of fully actuated individual robots i with dynamic model given by $\dot{\mathbf{q}}_i = \mathbf{v}_i$, $\dot{\mathbf{v}}_i = \mathbf{u}_i$, where $\mathbf{q}_i = [x_i, y_i]^T$ is the pose of robot i, $\mathbf{v}_i$ its velocity and $\mathbf{u}_i$ its control input. Robots are assembled together into a set Γ of groups, where each group j is modeled by a pair (g_j, p_j) that comprises the group pose and shape, respectively. As we are only considering robots whose configuration space can be represented in $\mathbb{R}^2$, then $g_j \in SE(2)$. Thus, we can define the abstraction a_j by:

$$\begin{aligned} a_j &= (g_j, p_j) \\ g_j &= (\mu_j^x, \mu_j^y, \theta_j) \\ p_j &= (s_j, t_j). \end{aligned} \tag{1}$$

This abstraction can be seen as an ellipse defined implicitly by $c_j(x,y) = 0$, centered at (μ_j^x, μ_j^y) with orientation θ_j and principal axis given by s_j and t_j.

Instead of using the controller defined in [1] to achieve grouping behavior, we chose to develop a simpler one considering a conventional artificial potential field approach, since this is the most common method in swarm navigation. This group behavior will be explained in the next section.

A. Grouping

A group $j \in \Gamma$ is formed by the robots which satisfy the constraint $c_j(\mathbf{q}_i) < 0$. Therefore, the curve $c_j(x,y) = 0$ can be seen as a border that limits and defines a group. To simplify the forthcoming equations, assume that $\mathbf{q}_i$ is given with respect to the orthonormal reference frame specified by the components of g_j.

Given a function $\phi(\mathbf{q}_i, a_j)$ that maps $\mathbf{q}_i$ to its radial distance to the border of group j, the normal function

$$f(\mathbf{q}_i, a_j) = e^{-\gamma\phi^2(\mathbf{q}_i, a_j)} \tag{2}$$

produces an artificial potential field, shown in Figure 1, whose maximum is located at the curve $c_j(x,y) = 0$, forming a bowl like surface, with γ being inversely proportional to the thickness of its walls. Based on this potential, the following control law is used:

$$\mathbf{u}_i = -k_1 \nabla f(\mathbf{q}_i, a_j) - k_2 \dot{\mathbf{q}}_i + \sum_{k \in N_i} F_r(\mathbf{q}_i, \mathbf{q}_k). \tag{3}$$

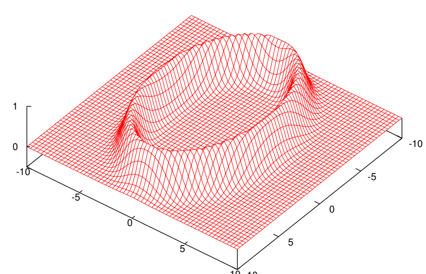


Fig. 1. Artificial potential field with $s_j = 5$, $t_j = 8$ and $\gamma = 1$.

Constants k_1 and k_2 are positive. The first term is a force that repels robots from the border of the group, assembling the agents that are on the inside and repelling those on the outside; the second term is a damping force, used to improve stability; and the third represents a local repulsive force, that prevents collisions among robots in a given neighborhood. The set N_i consists of every robot k that is within a certain distance limit δ from robot i. F_r is commonly defined in terms of $||\mathbf{q}_i - \mathbf{q}_k||$ and δ [8].

When considering that robots may be influenced by more than one group in the same workspace, the first term of (3) must become a summation that takes into account all elements of Γ. Also, it is necessary to apply some coordinate transformations on $\mathbf{q}_i$ between the global reference frame and the local ones, so that forces acting on robot i are in the same frame. Thus, we have

$$\mathbf{u}_i = -k_1 \sum_{j\in\Gamma} \psi_j^{-1}(\nabla f(\psi_j(\mathbf{q}_i), a_j)) - k_2\dot{\mathbf{q}}_i + \sum_{k\in N_i} F_r(\mathbf{q}_i, \mathbf{q}_k), \quad (4)$$

where ψ_j is the mentioned transformation and ψ_j^{-1} represents its inverse. In fact, the summation can be restricted only to groups within a neighborhood around group j since its robots will not be influenced by the potential of distant groups.

Equation (4) forces agents to avoid any intersection areas between groups, given an initial state where no robot lies on those areas. This behavior can be easily understood by observing that when a group collision begin to occur, the robots within a group are also repelled by the border of the other group. This feature will be explored to derive an algorithm for traffic control.

B. Congestion Avoidance

In order to move, rotate and reshape groups, simple linear controllers could be applied to g_j and p_j, respectively. These group controllers along with the ones defined by (4) establish a hierarchy in which robots are controlled implicitly according to the high level abstractions. Using this hierarchy, we can design control rules for a_j in order to avoid traffic congestions between large groups of robots.

The general idea of the avoidance algorithm is to take advantage of the geometric features of the virtual structure $c_j(x,y) = 0$ to create repulsive forces among groups in order to divert them from possible congestion areas.

Given two groups $m,n \in \Gamma$, let $\mathbf{p}$ be the centroid of the z intersection points between $c_m(x,y) = 0$ and $c_n(x,y) = 0$, $z \in \{1,2,3,4,\infty\}$. The repulsive force F_g between groups should be directly proportional to the penetration depth of $\mathbf{p}$ with reference to the ellipses. This depth can be computed by $\phi(\psi_m(\mathbf{p}), a_m)$, so we specify the force acting on group m as:

$$F_g(m,n) = \begin{cases} 0, & \text{for } z = 0 \\ \phi(\psi_m(\mathbf{p}), a_m)(\hat{\mathbf{g}}_m - \hat{\mathbf{p}}), & \text{for } z > 0, \end{cases} \quad (5)$$

where $\mathbf{g}_m = [\mu_m^x, \mu_m^y]^T$ and $\hat{\mathbf{p}}$ denotes the unit vector in the direction of $\mathbf{p}$.

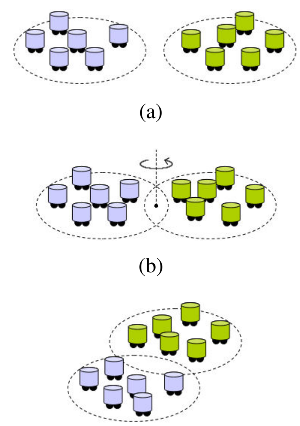
Equation (5) only tries to minimize the intersection area between the two groups involved, but it does not fulfill the purpose of deviating them. Therefore, it is also necessary to specify a rotational force R_g such that $F_g.R_g = 0$. Since (5) defines a vector field $F_g = (F_x, F_y)$, then we simply set $R_g = (-F_y, F_x)$ to satisfy the orthogonality constraint. Thus, the resulting force that deviates group m from n is

$$D(m,n) = F_g(m,n) + R_g(m,n). \quad (6)$$

Considering the dynamic model $\ddot{\mathbf{g}}_m = \mathbf{u}_m^g$ for the groups, we let the group controller be

$$\mathbf{u}_m^g = k_3 \sum_{n\in\Gamma} D(m,n) - k_4\dot{\mathbf{g}}_m + F_{goal}(\mathbf{g}_m), \quad (7)$$

where $F_{goal}(\mathbf{g}_m)$ is an attractive force that leads the group towards its goal position. The resulting behavior of (7) is shown in Figure 2. Note that it is necessary to make $R_g = 0$ if group m is near its goal, otherwise if $z > 0$ and more than one group tries to stop simultaneously at the same goal, an endless circular motion will occur, which is usually not a desired behavior.



(a)

(b)

(c)

Fig. 2. Execution steps of the congestion avoidance algorithm.

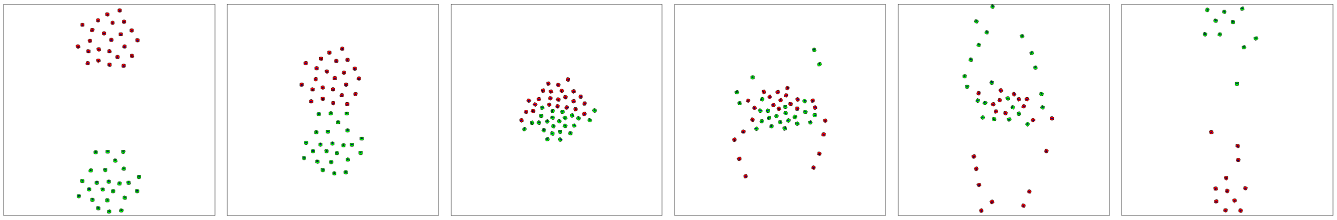
IV. EXPERIMENTS

To study the feasibility and performance of the proposed approach we executed a series of simulations and real experiments. For the simulations, we used Player/Stage [22], a well known framework for robot simulation and programming. We simulated robots based on the *P2DX* model, a differential drive robot, equipped with lasers. Real experiments were performed with a dozen *e-puck* robots [23], a small-sized differential robot equipped with a ring of 8 IR sensors for proximity sensing and a group of LEDs for displaying robot status. A bluetooth wireless interface allows local communication among robots and also with a remote computer..

In order to control the differential drive robots, to achieve better group alignment, the group velocity ($\dot{\mathbf{g}}_j$) was added as a new term into (4) and this new controller was adapted to account for nonholonomic constraints following the approach presented in [24].

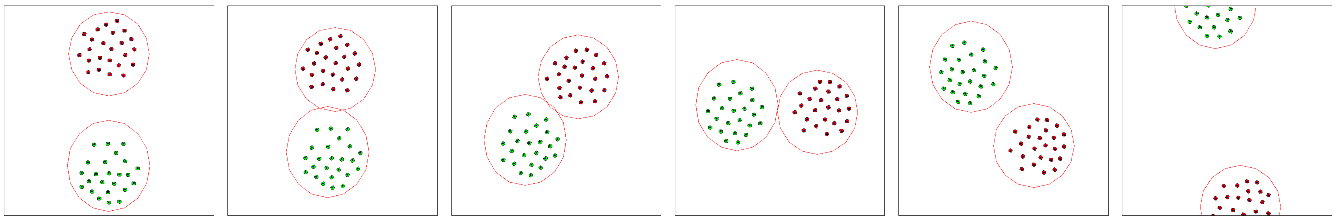
A. Simulations

Figure 4 presents the execution of the proposed algorithm in a simulated environment consisting of 48 virtual robots evenly divided into two groups that move in opposite directions. Also, in Figure 3, the same simulation is depicted,



(a) (b) (c) (d) (e) (f)

Fig. 3. Simulated execution using only local repulsive forces with two groups.



(a) (b) (c) (d) (e) (f)

Fig. 4. Simulated execution of the hierarchical coordination algorithm with two groups. The large polygon around the groups represent the abstraction.

but only local repulsive forces were used to avoid collisions among robots.

Through a visual inspection, it is easy to see that the robot movement is smoother and more stable using the proposed algorithm. Both groups behave more cohesively during navigation and a congestion scenario was completely avoided. On the other hand, the simulation that relied only on local repulsive forces for robot collision avoidance showed a large congestion when both groups encountered and mixed in a non-cohesive way.

A series of simulations was performed to evaluate the efficiency and the scalability of the proposed algorithm. For a varying number of robots we measured the execution time. Basically, for every fixed number of robots we ran the simulations 10 times and computed the arithmetic mean and the standard deviation of the results. Agents were divided evenly into groups, being positioned according to a normal distribution into a pre-specified area, and the initial configuration state of a_j was computed according to [1]. The number of iterations necessary for the last robot to overpass a fixed point in its initial movement direction was used as a measure of time. The results are shown in Figure 5.

As can be observed, the execution time of both algorithms rise monotonically as the number of robots increases. However, our approach has a much better performance saving a significant amount of time. For a hundred robots for example, the proposed algorithm shows average gains of 67% when compared to the execution without coordination. Also, our approach scales well as the groups become larger, which can be noted by the linear tendency and small slope of the dashed curve. Moreover, using the hierarchical traffic control, the general behavior of the groups has become more predictable due to the increase in stability and group coherence. This can be noted by the small standard deviation when running the algorithm several times.

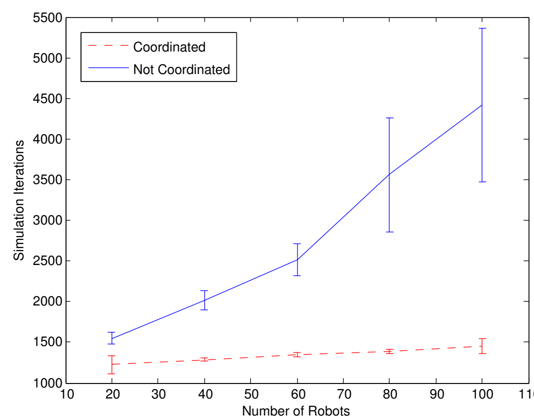
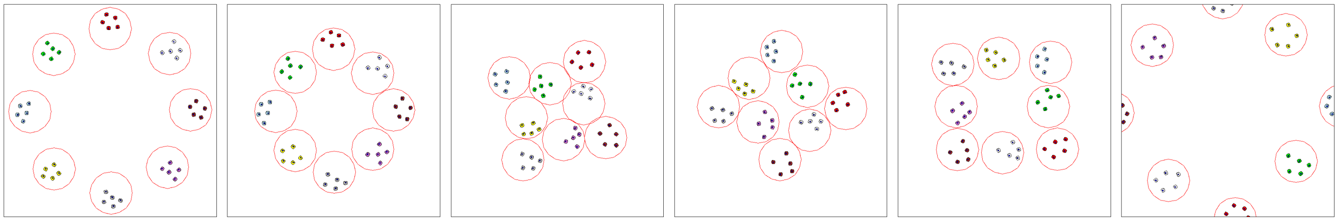


Fig. 5. Execution time taken by the algorithms; the bars represent the standard deviation.

Simulated experiments with more than two groups were also performed. Figure 6 shows an experiment with eight groups navigating in the environment. It can be observed that the algorithm is able to successfully avoid congestion when the eight groups pass through the same area. Results show average gains of 51% and 30% with forty robots evenly divided into four and eight groups, respectively. For more than two groups, we realized that a smaller convergence time can be achieved if they have similar shapes. An optimal case is when groups are modeled as circles as in Figure 6, since the behavior of the system becomes more symmetric.

B. Real Robots

The real experiments were conducted indoors using twelve *e-puck* robots (Figure 7). These experiments are important to show the feasibility of the algorithm in real scenarios, where all uncertainties caused by sensing and actuation errors may have an important role on the results.



(a) (b) (c) (d) (e) (f)

Fig. 6. Simulated execution of the hierarchical coordination algorithm with eight groups.

In these experiments, we used a swarm localization framework based on three overhead cameras and fiduciary markers for estimating robot pose and orientation. Since the *e-puck* is not designed to run at high speeds, for simplicity, a kinematic model $\dot{\mathbf{q}}_i = \mathbf{u}_i$ was used instead of the proposed dynamic model. Equation (4) was modified accordingly, by removing the damping force. Also, as the *e-puck*'s IR sensors have a very small range, we implemented a virtual sensor based on the localization system to detect neighboring agents.

Fig. 7. Twelve e-puck robots used in the experiments.

Figures 8 and 9 show snapshots from an execution with two and four groups, respectively. These proof of concept experiments indicate that the algorithm can work well to coordinate a swarm of robots, allowing them to navigate while avoiding congestions in an efficient way.

C. Discussion

One of the main advantages of this hierarchical approach is that it allows congestion avoidance without the need of specifically controlling the individual robots: by controlling the high level abstractions, robots are automatically deviated from the congestion points. It allows much more flexibility than our previous approach [20], which required certain maneuvers that restricted its applicability to few groups and specific trajectories. Also, the avoidance algorithm only requires feedback from the abstract state a_j, being independent of the grouping algorithm. Thus, other controllers may be used to ensure the shape of the swarm.

As in [1] and [13], some assumptions were made regarding the information available to the robots and abstractions. We consider that robot i has access to the state a_j of the group it belongs and knows its pose $\mathbf{q}_i$ in the group's local frame. This can be obtained, for example, through the use of a small group of aerial robots for controlling the abstractions and communicating with ground agents, as proposed in [13], or using consensus algorithms inside the groups.

One of the downsides of the grouping algorithm is the necessity of carefully tuning the constants. Values must be adjusted so that the virtual ellipses respect the speed and velocity constraints of the ground robots. Also, constants k_1 and γ must be adjusted according to the repulsion forces between agents, so that the summation of F_r along the set N_i does not cause robot i to leave the group. Moreover, the ϕ function explicitly defines the group shape, so it can be modified to consider other types of structures, such as a box or a triangle. Regarding the avoidance algorithm, constants k_3 and k_4 must be tuned in order to ensure that the abstraction does not surpass robots during its movement.

When we start to consider the nonholonomic constraints for differential drive robots, the ellipses should slow down or even stop when a robot is located very close to its border, thus allowing the agent time to maneuver (a similar behavior is noted in [6]). Unfortunately, this approach can lead to a deadlock state if the repulsive forces among robots take them to fully expand inside the ellipse. Also, in real scenarios, measurement errors can lead to failures when robots are close to the group borders.

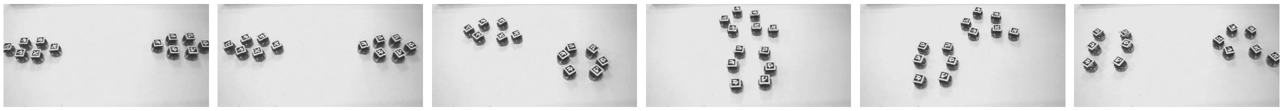
Experimental results showed that performance gains tend to decrease when considering a large number of small groups, compared to few large groups. In these scenarios, a new hierarchical level could be implemented by defining groups as single agents that are controlled by an even higher level abstraction.

Another interesting point is that the constant k_3, which multiplies the group deviation force, can be modified into a function that considers the area or the number of robots in a group, in a way that a simple priority system can emerge. For example, it is possible that a group with many robots continue on a straight path while smaller groups deviate from it by applying this technique. This behavior has not been explored yet, but may lead to some interesting results.

V. CONCLUSION

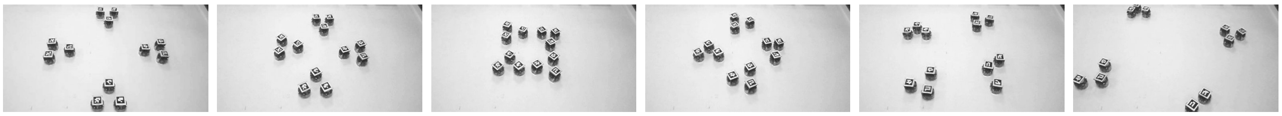
In this work we proposed an algorithm for controlling the traffic of a robotic swarm with the objective of avoiding congestion situations in scenarios where various groups simultaneously cross the same area in the environment. It has put together two distinct topics that are widely studied in mobile robotics: grouping behaviors for robotic swarms and traffic control.

The algorithm uses a high level abstraction, designed using an artificial potential field approach, that groups robots in a hierarchical fashion. In this way, individual robots are controlled implicitly by changing the abstraction parameters. By considering the geometrical features of the abstraction,



(a) (b) (c) (d) (e) (f)

Fig. 8. Real execution of the hierarchical coordination algorithm with two groups.



(a) (b) (c) (d) (e) (f)

Fig. 9. Real execution of the hierarchical coordination algorithm with four groups.

which were modeled as ellipses in this paper, we developed an avoidance algorithm that relied on virtual forces generated from the intersection between groups. Several experiments were performed both in simulated and in real scenarios, which demonstrated that the proposed algorithm improved navigation efficiency by completely avoiding congestions.

Despite the good results obtained by the hierarchical algorithm, there is still room for improvement. For example, the geometric features of the groups, as well as their possible behaviors, were not exploited to its maximum potential, as expansions and contractions were not considered. This is an interesting feature, since it can be merged with a path planner to allow a similar traffic control algorithm to work in environments with obstacles. Moreover, we would like to see experiments using different shapes to model the swarm as well as other controllers for the ground agents. Also, experiments using real aerial robots would be an interesting addition. We believe that by further exploring these new possibilities we can achieve a more robust and efficient hierarchical navigation system for robotic swarms.

REFERENCES

[1] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 865–875, Oct. 2004.

[2] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," in *Computer Graphics*, 1987, pp. 25–34.

[3] J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, no. 3, pp. 171–194, 1999.

[4] K.-H. Tan and M. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Nov. 1996, pp. 132–139.

[5] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, Dec. 2001.

[6] A. Kamphuis and M. Overmars, "Motion planning for coherent groups of entities," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, May 2004, pp. 3815–3822.

[7] L. Barnes, M. Fields, and K. Valavanis, "Swarm formation control utilizing elliptical surfaces and limiting functions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1434–1445, Dec. 2009.

[8] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, June 2005.

[9] T.-Y. Li and H.-C. Chou, "Motion planning for a crowd of robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, Sept. 2003, pp. 4215–4221.

[10] A. Kamphuis and M. H. Overmars, "Finding paths for coherent groups using clearance," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2004, pp. 19–28.

[11] N. Michael, C. Belta, and V. Kumar, "Controlling three dimensional swarms of robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2006, pp. 964–969.

[12] S. Hou, C. Cheah, and J. Slotine, "Dynamic region following formation control for a swarm of robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2009, pp. 1929–1934.

[13] L. Chaimowicz and V. Kumar, "Aerial shepherds: Coordination among uavs and swarms of robots," in *7th International Symposium on Distributed Autonomous Robotic Systems*, June 2004.

[14] Y. Cao, A. Fukunaga, A. Kahng, and F. Meng, "Cooperative mobile robotics: antecedents and directions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Aug. 1995, pp. 226–234.

[15] D. Grossman, "Traffic control of multiple robot vehicles," *IEEE Journal of Robotics and Automation*, vol. 4, no. 5, pp. 491–497, Oct. 1988.

[16] S. Kato, S. Nishiyama, and J. Takeno, "Coordinating mobile robots by applying traffic rules," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, July 1992, pp. 1535–1541.

[17] K. Dresner and P. Stone, "Multiagent traffic management: An improved intersection control mechanism," in *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM Press, July 2005.

[18] K. Viswanath and M. Krishna, "Sensor network mediated multi robotic traffic control in indoor environments," in *Proceedings of the International Conference on Advanced Robotics*, Oct. 2007.

[19] L. Marcolino and L. Chaimowicz, "Traffic control for a swarm of robots: Avoiding target congestion," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 1955–1961.

[20] ——, "Traffic control for a swarm of robots: Avoiding group conflicts," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 1949–1954.

[21] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple nonholonomic robots," in *Proceedings of the 10th International Symposium on Distributed Autonomous Robotic Systems*, Nov. 2010.

[22] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proceedings of the 11th International Conference on Advanced Robotics*, 2003, pp. 317–323.

[23] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, vol. 1, 2009, pp. 59–65.

[24] A. De Luca and G. Oriolo, "Local incremental planning for nonholonomic mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, May 1994, pp. 104–110.