On Segregative Behaviors Using Flocking and Velocity Obstacles

Vinicius Graciano Santos, Mario F. M. Campos, and Luiz Chaimowicz

Abstract This paper presents a novel approach to swarm navigation that combines hierarchical abstractions, flocking behaviors, and an efficient collision avoidance mechanism. Our main objective is to keep large groups of robots segregated while safely navigating in a shared environment. For this, we propose the *Virtual Group Velocity Obstacle*, which is an extension of the Velocity Obstacle concept for groups of robots. By augmenting velocity obstacles with flocking behaviors and hierarchical abstractions, we are able to navigate robotic swarms in a cohesive and smooth fashion. A series of simulations and real experiments were performed and the results show the effectiveness of the proposed approach.

1 Introduction

The use of large groups of robots in the execution of complex tasks has received much attention in recent years. Generally called *Robotic Swarms*, these systems employ a large number of simpler agents to perform different types of tasks, oftentimes inspired by their biological counterparts.

A basic requirement for most robotic swarms is the ability for safe navigation in shared environments, *i.e.*, the ability of moving to specific goals while avoiding collisions with obstacles, teammates, and other groups. A desired property in this case is to keep robots close to their kins and avoid merging with other groups. This is called segregation and, as discussed previously [17], is a phenomenon that is seen in several biological systems.

This paper presents a novel approach to swarm navigation that keeps large groups of robots segregated while safely navigating in a shared environment. Our approach consists of extending the concept of Velocity Obstacles [10] with flocking behaviors

Authors are with the Vision and Robotics Laboratory (VeRLab), Computer Science Department, Universidade Federal de Minas Gerais, Brazil. e-mails: {vgs, mario, chaimo}@dcc.ufmg.br

and hierarchical abstractions. Basically, Velocity Obstacles define the set of robot velocities that would result in a collision between the robot and an obstacle moving at a given velocity. We first augment this concept with flocking behaviors [23], which enables different robot groups to avoid collisions and navigate in a cohesive fashion using global perception. But when we restrict the robot's perception to a smaller area, this approach is unable to keep groups segregated. To solve this problem, we employ hierarchical abstractions [12]: we consider a group of robots as a single entity and make individual robots avoid velocities that would result in collisions with this entity, *i.e.*, we prevent robots from mingling with other groups.

We call this approach *Virtual Group Velocity Obstacles* and we present a series of experiments that show its capability of attaining our goal of safely navigating large robotic groups while keeping them segregated.

This paper is organized as follows: Section 2 discusses related work in the field of collision avoidance, swarm navigation, and hierarchical abstractions. The Velocity Obstacle concept is presented in Section 3. Section 4 introduces our methodology used to develop the Virtual Group Velocity Obstacle. Experimental results in simulated and real environments are presented in Section 5, while Section 6 brings the conclusion and directions for future work.

2 Related Work

One of the earliest works that considered the problem of controlling a swarm of agents was presented by Reynolds [23] with the aim of realistically simulating a flock of birds, known as *boids*. In that work, local interactions among agents within a neighboring area define an emergent behavior for the whole flock. Such interactions can be modeled as a special case of the social potential field method [22], an extension of the classical artificial potential field technique [16] that specifically deals with multi-agent systems.

The artificial potential field approach has been widely employed for controlling multi-robot and swarm systems [24]. Moreover, many works have focused on its use together with flocking principles in order to obtain specific behaviors, such as area coverage [15], moving in formation [3], converging into shapes [8], shepherding [18], segregation [17], and so on. However, it is known that the method is not oscillation-free and suffers from local minima, which is an intrinsic property that can arise from the combination of potentials, specially in unknown environments.

Recent work on robot collision avoidance has provided methods that are guaranteed to be collision-free and oscillation-free [5, 14, 25], even under nonholonomic constraints [2]. These methods rely on the concept of Velocity Obstacles [10], which is an extension of the Configuration Space Obstacle [19] for a time-varying system. A Velocity Obstacle defines the set of robot velocities that would result in a collision between the robot and an obstacle moving at a given velocity. Thus, the robot performs an avoidance maneuver at a specific time by selecting velocities that do not belong to that set. This approach has been widely used and extended for multi-agent navigation [1, 5, 6, 14, 25], even when considering uncertainties in position, shape, and velocity of the obstacles [11, 25]. An important extension was the development of the Reciprocal Velocity Obstacle [6], which acknowledged that most works on collision avoidance have not taken into account that obstacles' motion may be affected by the presence of the agent. That is, they have overlooked the reciprocity that arises when those obstacles are in fact other agents that can also react according to the robot's behavior, which could lead to oscillations in the system.

A different hierarchical paradigm considers the whole group as a single entity, sometimes called virtual structure [26], which effectively reduces the dimensionality of the control problem. The desired behavior is assigned to this structure, which implicitly controls the robotic swarm. 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 hierarchical abstraction that allows decoupled control of the pose and shape of a team of robots, was developed in [4]. This work was extended in [21], to account for three dimensional swarms. Based on the former, a hierarchical cooperation mechanism between multiple unmanned aerial and ground vehicles was developed [7], where UAVs are responsible for estimating the configuration of the ground robots and for sending control messages to the groups.

In our previous work [12], we have shown that hierarchical abstraction paradigms can be used in conjunction with simple collision avoidance techniques in order to achieve complex behaviors for swarm systems. We proposed a mechanism that allows large groups to deviate from each other during navigation and we have applied it to the problem of traffic control. In the present paper, we propose a different approach. We take advantage of the robust collision avoidance techniques that have been developed [6] and use them in conjunction with simple flocking rules and hierarchical abstractions for swarm navigation. Specifically, our goal is to ensure greater cohesion between robotic groups that navigate in the same workspace. That is, robots in the same group must stay together while avoiding merging with distinct groups. Henceforth, we denote this desired property as a *segregative behavior*.

3 Velocity Obstacles

Let A and B be two robots moving on the plane. Each robot i is fully actuated with kinematic model given by $\dot{\mathbf{p}}_i = \mathbf{v}_i$, where $\mathbf{p}_i = [x_i, y_i]^T$ is its pose and $\mathbf{v}_i$ its velocity. The velocity obstacle $VO^A_B(\mathbf{v}_B)$ of B to A is defined as the set of all velocities $\mathbf{v}_A$ that will result in a collision among robots A and B at some instant in time [10]. More precisely, we define $\lambda(\mathbf{p}, \mathbf{v})$ as a ray starting at $\mathbf{p}$ heading in the direction of $\mathbf{v}$ and $B \oplus -A$ as the Minkowski sum of B and $-A$, where $-A$ represents robot A reflected about its reference point.

$$\lambda(\mathbf{p}, \mathbf{v}) = \{\mathbf{p} + t\mathbf{v} \,|\, t \geq 0\} \tag{1}$$

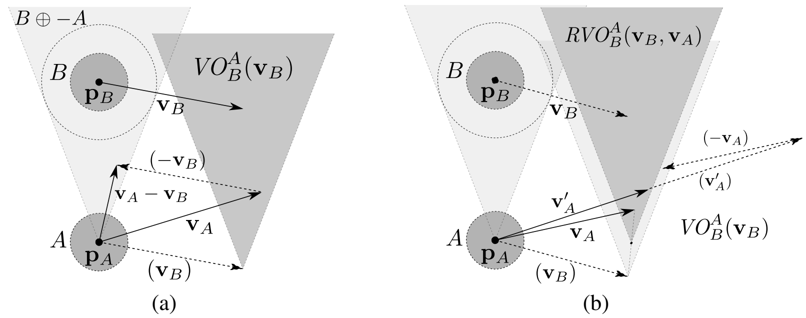


Fig. 1 (a) The Velocity Obstacle $VO_B^A(\mathbf{v}_B)$. (b) The Reciprocal Velocity Obstacle $RVO_B^A(\mathbf{v}_B, \mathbf{v}_a)$.

Given these definitions, we can say that a velocity $\mathbf{v}_A \in VO_B^A(\mathbf{v}_B)$ if and only if the ray starting at $\mathbf{p}_A$ heading in the direction $\mathbf{v}_A - \mathbf{v}_B$ intersects $B \oplus -A$. Therefore, the full set of velocities that specifies a Velocity Obstacle can be denoted as:

$$VO_B^A(\mathbf{v_B}) = \{\mathbf{v}_A \,|\, \lambda(\mathbf{p}_A, \mathbf{v}_A - \mathbf{v}_B) \cap (B \oplus -A) \neq \emptyset\}. \tag{2}$$

This set has an interesting property: if A selects a velocity that is outside its Velocity Obstacle induced by B, and if B maintains its current velocity, it is guaranteed that a collision between them will not occur [10]. Figure 1(a) shows an example of a Velocity Obstacle in a system with two circular robots. As it can be seen, $VO_B^A(\mathbf{v}_B)$ is a cone with its apex at $(\mathbf{v}_B)$.

When dealing with multi-robot systems, navigation based on the original Velocity Obstacle suffers from oscillation issues. In order to overcome this problem, a new velocity is chosen such that it is the average of the robot's current velocity and a velocity that lies outside the Velocity Obstacle [6]. Formally, the Reciprocal Velocity Obstacle is defined as:

$$RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) = \{\mathbf{v}'_A \,|\, 2\mathbf{v}'_A - \mathbf{v}_A \in VO_B^A(\mathbf{v}_B)\}. \tag{3}$$

This new set contains all velocities that are the average of $\mathbf{v}_A$ and a velocity within $VO_B^A(\mathbf{v}_B)$, which can be seen as the cone $VO_B^A(\mathbf{v}_B)$ translated such that its apex lies at the mean of $\mathbf{v}_A$ and $\mathbf{v}_B$, as shown in Figure 1(b). Selecting the velocity which is closest to the robot's prior velocity and that also lies outside the set $RVO_B^A(\mathbf{v}_B, \mathbf{v}_A)$, guarantees a collision-free and oscillation-free navigation between a pair of robots [6].

Finally, in order to select inputs when dealing with Velocity Obstacles, an optimization problem must be solved. Several different approaches have been proposed in [5, 6, 10, 14, 25]. In the following section, we discuss how to couple flocking behaviors with sampling-based velocity selection [6].

4 Methodology

As mentioned, our main objective is to safely navigate large groups of robots in a shared environment while maintaining segregation among groups. Our approach consists of extending the concept of Reciprocal Velocity Obstacles with flocking behaviors and hierarchical abstractions.

We consider that robots are assembled together into a set of groups $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \ldots \cup \Gamma_N$, where $\forall j,k: j \neq k \rightarrow \Gamma_j \cap \Gamma_k = \emptyset$. Let $\Phi_k \subseteq \Gamma_k$ be the set of robots belonging to group Γ_k that are within the neighborhood N_i, thus being perceived by robot i; and $\mathbf{p}(\Phi_k)$, $\mathbf{v}(\Phi_k)$ be the average position and velocity of group Φ_k, respectively.

Flocking behaviors can be achieved by a set of simple rules defined among neighboring robots [23]. Usually, robot controllers are derived by giving feedback on errors, such as relative velocities and relative positions. In our approach, we extend the velocity selection process presented in [6], which fast samples the set of admissible velocities and selects the best one according to an utility function.

Let $\mathbf{v}_i^{\text{pref}}$ be the preferred velocity of robot i, such as the vector pointing at the goal's direction with magnitude that is equal to the maximum allowed speed. In each iteration, velocities are sampled using a uniform distribution from the set of admissible velocities $AV^i(\mathbf{v}_i)$, that comprises the possible new velocities given the kinematic and dynamic constraints:

$$AV^i(\mathbf{v}_i) = \{\mathbf{v}_i' \mid \|\mathbf{v}_i'\| < v_i^{max} \wedge \|\mathbf{v}_i' - \mathbf{v}_i\| < a_i^{max} \Delta t\}, \tag{4}$$

where v_i^{max} and a_i^{max} are the maximum speed and maximum acceleration of robot i, respectively, and Δt is the time step of the system.

Ideally, the selected velocity $\mathbf{v}_i^{\text{new}}$ among the sampled set should lie outside the union of all RVOs generated by other robots and VOs generated by dynamic and static obstacles. However, as the environment may become crowded to the point that no admissible velocities will exist, the algorithm is allowed to selected a velocity that belongs to the generated set of Velocity Obstacles, but it is penalized by this choice according to the following function:

$$\mathbf{v}_i^{\text{flock}} = \mathbf{v}_i^{\text{pref}} + \alpha(\mathbf{v}(\Phi_k) - \mathbf{v}_i) + \beta(\mathbf{p}(\Phi_k) - \mathbf{p}_i) \tag{5}$$

$$P_i(\mathbf{v}_i') = \frac{w}{c_i(\mathbf{v}_i')} + \|\mathbf{v}_i^{\text{flock}} - \mathbf{v}_i'\|, \tag{6}$$

with $i \in \Gamma_k$, where α rules the convergence of the robot's current velocity to its neighbors' average velocity, β is the weight that governs the behavior which makes robots move toward the centroid of their neighboring agents, and w regulates the avoidance behavior between sluggish and aggressive. Function $c_i(\mathbf{v}_i')$ is the expected time to collision, which is computed by solving the set of ray intersection equations induced by (2), (3) and taking their minimum. Note that, by setting $\alpha = \beta = 0$, the selection approach is reduced to the original RVO method [6]. Thus, we select a new velocity $\mathbf{v}_i^{\text{new}}$ that minimizes the penalty function P_i over the sampled set of velocities $S \subseteq AV^i(\mathbf{v}_i)$. Figure 2 exemplifies this sampling-based selection procedure.

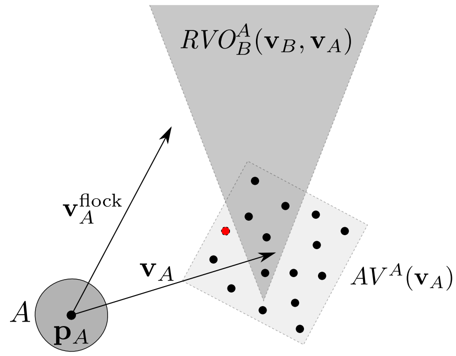
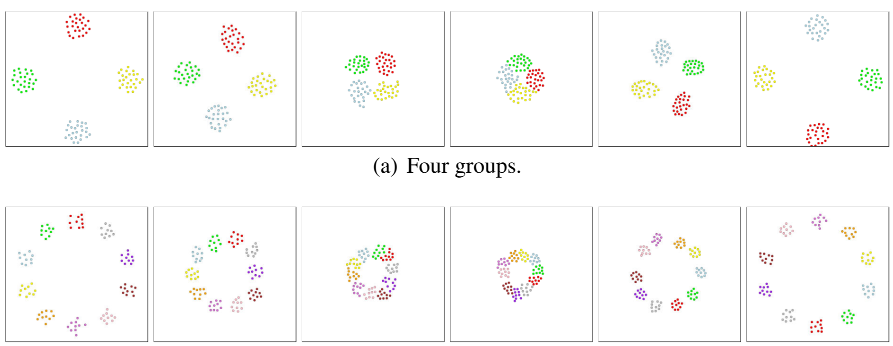


Fig. 2 Sampling-based velocity selection. Admissible velocities which were sampled are represented by small circles. The red sample is chosen as it minimizes the penalty function.

$$\mathbf{v}_i^{\text{new}} = \underset{\mathbf{v}_i' \in S}{\operatorname{argmin}} P_i(\mathbf{v}_i') \tag{7}$$

Equation (5) is responsible for the flocking behavior, which can be achieved by fine tuning its constants. For instance, there are triples of (α, β, w) that can lead robots into tightly aggregated groups. In these scenarios, groups moving in opposite directions show an emerging segregative behavior. That is, they tend to smoothly avoid each other without merging, as seen in Figure 3.



(a) Four groups.

(b) Ten groups.

Fig. 3 Simulated execution of the flocking algorithm with distinct groups moving in opposite directions using global sensing.

As groups move toward each other, robots will try to select velocities that lie outside of the combined RVOs induced by the other group. Since this group is tightly packed, robots will select velocities that may lead them to avoid the group as a whole. Given the reciprocity, the other group will select its velocities accordingly.

Nevertheless, the same behavior cannot be guaranteed when dealing with a small sensing neighborhood, since robots will not be able to predict incoming groups. In this case, merging can occur if the robot is not able to correctly maneuver given its current velocity. To account for this problem, we introduce a virtual Velocity Obstacle that is responsible for blocking velocities which may lead to merging among groups. Specifically, we denote this virtual obstacle as the *Virtual Group Velocity Obstacle* (VGVO), which is shown in Figure 4.

The concept of VGVO is simple: each robot i senses the presence of every robot j within a neighborhood N_i and builds the shape of each group of robots, but of its own group. These shapes are considered as virtual obstacles in the workspace of robot i that move with the average velocity of the respective group that has been used in the building process. Hence, a virtual Velocity Obstacle can be built for each shape in order to define the set of velocities that would lead the robot to merge with a different group if the latter maintains its current average velocity.

Let $R(\mathbf{p}_i)$ denote all the points that represent robot i in its workspace. Consequently, the Virtual Group Velocity Obstacle of robot i induced by group Φ_k can be defined as:

$$VGVO^i_{\Phi_k}(\mathbf{v}(\Phi_k)) = \{\mathbf{v}'_i \,|\, \lambda(\mathbf{p}_i, \mathbf{v}'_i - \mathbf{v}(\Phi_k)) \cap C(\mathbf{p}_i, \Phi_k) \neq \emptyset\}, \tag{8}$$

$$C(\mathbf{p}_i, \Phi_k) = \mathrm{Shape}(\bigcup_{j \in \Phi_k} R(\mathbf{p}_j)) \oplus -R(\mathbf{p}_i), \tag{9}$$

where $\mathrm{Shape}(Q)$ is the shape of the set of points Q, which could be represented as the smallest enclosing disc, the convex hull or the more general class of α-shapes [9].

Equation (9) refers to the idea of the hierarchical abstraction paradigm presented in Section 2, in which the whole group is considered as a single entity. In every iteration, groups are abstracted into single entities that move according to their average velocities. In this way, single robots navigate using the RVO in conjunction with the VGVO, which guarantees a collision-free navigation among groups while maintaining segregation. Furthermore, we can assume that groups will deviate from single robots given the reciprocity of the RVO. Therefore, we can also specify a Virtual Group Reciprocal Velocity Obstacle with the following definition:

$$VGRVO^i_{\Phi_k}(\mathbf{v}(\Phi_k), \mathbf{v}_i) = \{\mathbf{v}''_i \,|\, 2\mathbf{v}''_i - \mathbf{v}_i \in VGVO^i_{\Phi_k}(\mathbf{v}(\Phi_k))\}. \tag{10}$$

5 Experiments

To study the feasibility of the proposed approach, we executed a series of simulations and real experiments[1]. For the simulations, we used Player/Stage [13], a well known framework for robot programming and simulation. Real experiments were performed with a dozen *e-puck* robots, a small-sized differential robot equipped with

[1] a video showing the experiments is available at: http://www.youtube.com/watch?v=ylYMbwOduqg

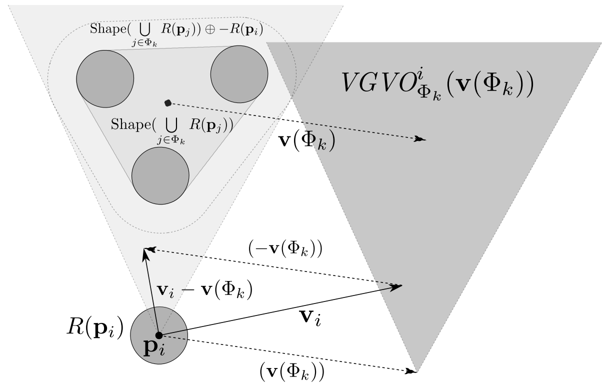


Fig. 4 The Virtual Group Velocity Obstacle $VGVO^i_{\Phi_k}(\mathbf{v}(\Phi_k))$.

a ring of 8 IR sensors for proximity sensing and a DS-PIC processor. A bluetooth wireless interface allows local communication among robots along with a remote computer.

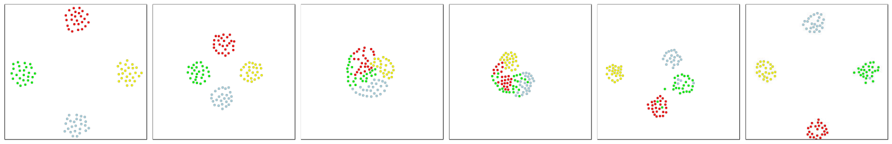
In all experiments, constants α, β, and w were set manually. When comparing the behavior of different algorithms, these constants remain the same between executions. Additionally, the neighborhood area is defined as five times the robot's radius in experiments with restricted sensing. Furthermore, we assume that each robot has the ability to map every other robot to its respective group.

5.1 Simulations

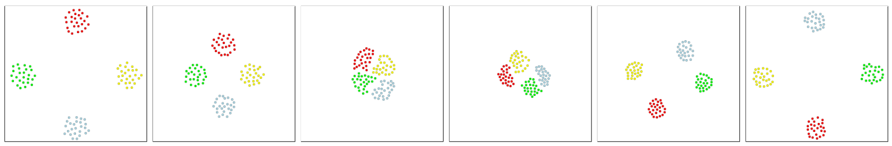
Figure 3 presents the execution of the flocking algorithm in a simulated environment consisting of 100 virtual robots evenly distributed into four and ten groups that move in opposite directions. Both simulations rely on global sensing, thus robots can perceive each other regardless of their relative distances. Through visual inspection we can see that, robots tend to remain segregated into homogeneous groups using our approach.

When we restrict robot perception to a local sensing, the flocking approach alone does not guarantee segregation, as shown in Figure 5(a). On the other hand, as depicted in Figure 5(b), the VGRVO is able to keep robots segregated even within a local sensing scenario. In this case, the virtual obstacle was built upon the groups' α-shape [9]. Thus, the addition of VGRVOs has led to a successful navigation while maintaining the desired segregation property.

In a recent work [17], a formal way of measuring segregation among groups of agents has been proposed. Two groups Γ_A and Γ_B are said to be segregated if the average distance between robots in the same group is less than the average distance between robots in distinct groups. Thus, the following restriction must hold



(a) Flocking approach alone does not guarantee segregation.



(b) Virtual Group Reciprocal Velocity Obstacle.

Fig. 5 Simulated execution of the segregative behavior algorithm with distinct groups moving in opposite directions using local sensing.

$$(d_{\text{avg}}^{AA} < d_{\text{avg}}^{AB}) \wedge (d_{\text{avg}}^{BB} < d_{\text{avg}}^{AB}), \tag{11}$$

where d_{avg}^{AB} is the average distance between robots in group Γ_A and Γ_B.

Figure 6 depicts a comparative analysis about the emerging segregative behaviors using the proposed algorithms. For these simulations, a scenario consisting of 200 virtual robots evenly divided into two distinct groups was used. Initially, agents were randomly positioned into a circular area according to a normal distribution. After that, groups were ordered to swap their positions. Figures 6(a) and 6(b) show the results for the original RVO approach and our flocking extension, respectively, while relying on global sensing capabilities for all robots. As it can be seen, constraint (11) holds true for all time slices of the second simulation, which indicates that a segregative behavior has emerged. On the other hand, since the RVO algorithm tends to form merged lanes of moving robots during navigation, (11) will not hold, as expected. This can be confirmed visually in Figure 6(a), as there exists intersection points between the curve d_{avg}^{AB} and one of the others. Figures 6(c) and 6(d) compare our flocking algorithm without and with the use of VGRVO, respectively. In these simulations, robots had their sensors capabilities constrained into a small local neighborhood when compared to the size of their own group. Due to this restriction, groups were not able to remain segregated during navigation by means of flocking only, since (11) did not hold for all time steps in Figure 6(c). However, the use of VGRVOs under these conditions has shown to be effective for maintaining a segregative behavior throughout the simulation, as shown in Figure 6(d).

Although our flocking mechanism can improve group navigation in terms of cohesion and symmetry, specially when dealing with multiple groups, such as in Figure 3(b), the system's performance in terms of elapsed time may decrease. For instance, in Figure 6, when curve d_{avg}^{AB} reaches again its initial value, it means that both groups have swapped their average positions. Hence, it is easy to see that robots using the original RVO approach have reached their goals faster than when applying flocking. Intuitively, the loss in performance happens because the second and third

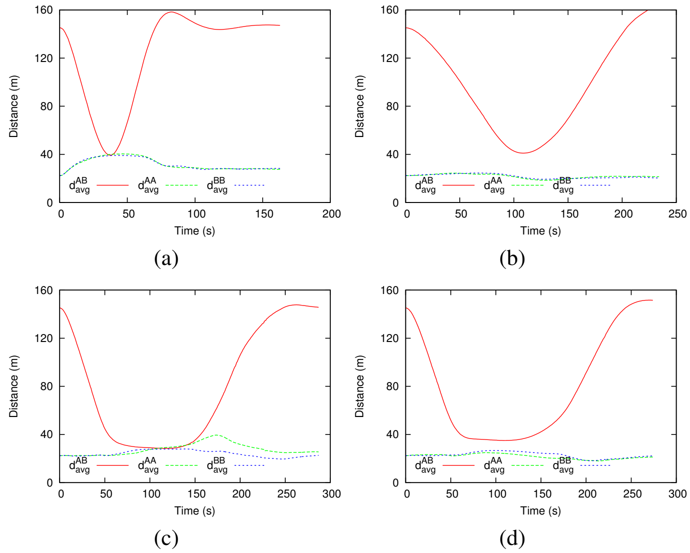


Fig. 6 Segregative behavior analysis for 200 robots evenly distributed into two groups that swap positions. (a) RVO with global sensing. (b) Flocking with global sensing. (c) Flocking with local sensing. (d) VGRVO with local sensing.

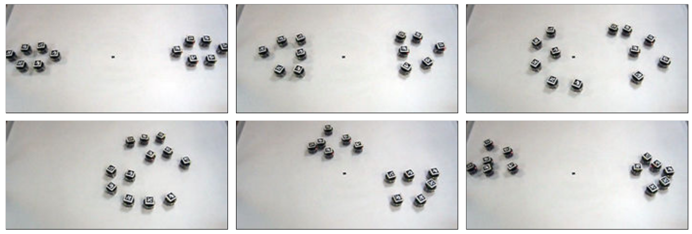
terms of equation (5) may play a damping role. Furthermore, as neighbors get closer to each other, robots will select safer velocities in order to avoid collisions, while trying to maintain the flock. Thus, slower speeds will have higher priority during the selection process.

5.2 Real Robots

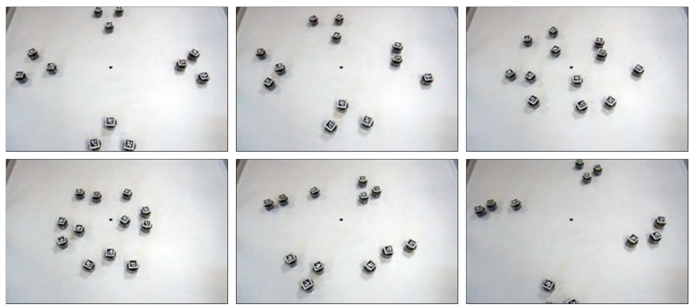
Real experiments were conducted indoors using twelve *e-puck* robots. These experiments are important in order to show the feasibility of the algorithm in real scenarios, where all uncertainties caused by sensing and actuation errors may have an important role on the results.

In these experiments, we used a swarm localization framework based on an overhead camera and fiduciary markers for estimating robot's pose and orientation. 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. In order to control the differential drive robots, a damping term based on the current velocity was added to (4) in order to achieve better stability. To account for nonholonomic constraints, input velocities were transformed following the approach presented in [20].

Figure 7 shows snapshots from an execution of VGRVO with limited sensing with two and four groups. We can visually inspect that the behaviors obtained with



(a) Two robotic groups.



(b) Four robotic groups.

Fig. 7 Real execution of the VGRVO algorithm with different group sizes.

the real robots is pretty similar to the simulation results. In terms of segregation, the average distances for two groups follow the trend showed in Figure 6(d) for the simulated experiments: the average distance between robots in the same group is always less than the average distance between robots in distinct groups. These proof of concept experiments indicate that the algorithm can work well to coordinate groups of real robots, allowing them to navigate while maintaining a segregative behavior in an efficient way.

6 Conclusion

This paper presented an approach that ensures greater cohesion between robotic groups that navigate in shared environments. That is, robots flock together with their own group while remaining segregated from others. Our method does not rely on any communication to achieve coordination and works in a distributed fashion.

We achieve flocking behaviors for robotic swarms using Velocity Obstacles by choosing a proper utility function for selecting velocities. Results have shown that segregative behaviors have emerged during our experiments. We realized that this was not the case when dealing with restricted sensing capabilities. Therefore, we introduced a novel concept: the Virtual Group Velocity Obstacle, which is a virtual obstacle that is created in order to prevent aggregation between distinct robotic groups. The VGVO resembles ideas from the hierarchical abstraction paradigm, where groups are abstracted into single entities with the aim of reducing the dimensionality of the control problem. Several experiments were performed in simulated and real scenarios, which demonstrated the effectiveness of the proposed approach.

Despite the good results, there is still room for improvement. For instance, Velocity Obstacle algorithms are well known for allowing high-speed navigation. However, our algorithm tends to prioritize slower speeds in order to maintain stable relative distances and velocities among robots. Furthermore, it is worth noting that the use of VGVOs is only justifiable in constrained sensing conditions, since the cost of computing the virtual shape for each group can scale up given larger sensing neighborhoods. In addition, the conservative nature of the virtual obstacle may prevent feasible velocities from being chosen, which given a larger sensing radius may turn to be overly conservative. Finally, uncertainty models could be easily included following the approach presented in [11, 25].

Future work will investigate improvements along these lines, as well as experiments containing static and dynamic obstacles, other methods for selecting velocities, and different shape descriptors. We believe that velocity obstacle based algorithms have the potential of being easily manipulated in order to couple simple behavior rules. For example, by specifying velocity obstacles that block undesirable velocities, such as the VGVO for segregation. Further investigation may lead to interesting results that shall further extend the velocity obstacle framework.

Acknowledgements This work was partially supported by CNPq and Fapemig. We would like to thank Prof. Vijay Kumar for some insights on segregative behaviors.

References

1. Abe, Y., Yoshiki, M.: Collision avoidance method for multiple autonomous mobile agents by implicit cooperation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 1207 – 1212 (2001)
2. Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., Siegwart, R.: Optimal reciprocal collision avoidance for multiple non-holonomic robots. In: Proceedings of the 10th International Symposium on Distributed Autonomous Robotic Systems (2010)
3. Balch, T., Hybinette, M.: Social potentials for scalable multirobot formations. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 73–80 (2000)
4. Belta, C., Kumar, V.: Abstraction and control for groups of robots. IEEE Transactions on Robotics **20**(5), 865–875 (2004)
5. van den Berg, J., Guy, S.J., Lin, M.C., Manocha, D.: Reciprocal n-body collision avoidance. In: The 14th International Symposium on Robotics Research. Springer Tracts in Advanced Robotics (2009)
6. van den Berg, J., Lin, M., Manocha, D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In: IEEE International Conference on Robotics and Automation, pp. 1928 –1935 (2008)
7. Chaimowicz, L., Kumar, V.: Aerial shepherds: Coordination among uavs and swarms of robots. In: 7th International Symposium on Distributed Autonomous Robotic Systems (2004)

8. Chaimowicz, L., Michael, N., Kumar, V.: Controlling swarms of robots using interpolated implicit functions. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2498–2503 (2005)
9. Edelsbrunner, H., Kirkpatrick, D., Seidel, R.: On the shape of a set of points in the plane. IEEE Transactions on Information Theory **29**(4), 551 – 559 (1983)
10. Fiorini, P., Shillert, Z.: Motion planning in dynamic environments using velocity obstacles. International Journal of Robotics Research **17**, 760–772 (1998)
11. Fulgenzi, C., Spalanzani, A., Laugier, C.: Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid. In: IEEE International Conference on Robotics and Automation, pp. 1610 –1616 (2007)
12. G Santos, V., Chaimowicz, L.: Hierarchical congestion control for robotic swarms. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4372 –4377 (2011)
13. Gerkey, B.P., Vaughan, R.T., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: Proceedings of the 11th International Conference on Advanced Robotics, pp. 317–323 (2003)
14. Guy, S.J., Chhugani, J., Kim, C., Satish, N., Lin, M., Manocha, D., Dubey, P.: Clearpath: highly parallel collision avoidance for multi-agent simulation. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 177–187. ACM, New York, NY, USA (2009)
15. Howard, A., Mataric, M., Sukhatme, G.: Mobile sensor network deployment using potential fields: A distributd, scalable solution to the area coverage problem. In: Proceedings of the 6th International Symposium on Distributed Autonomous Robotic Systems (2002)
16. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 2, pp. 500 – 505 (1985). DOI 10.1109/ROBOT.1985.1087247
17. Kumar, M., Garg, D., Kumar, V.: Segregation of heterogeneous units in a swarm of robotic agents. IEEE Transactions on Automatic Control **55**(3), 743 –748 (2010)
18. Lien, J.M., Bayazit, O., Sowell, R., Rodriguez, S., Amato, N.: Shepherding behaviors. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4159–4164 (2004)
19. Lozano-Perez, T.: Spatial planning: A configuration space approach. IEEE Transactions on Computers **C-32**(2), 108 –120 (1983)
20. Luca, A.D., Oriolo, G., Vendittelli, M.: Stabilization of the unicycle via dynamic feedback linearization. In: 6th IFAC Symposium on Robot Control, pp. 397 – 402 (2000)
21. Michael, N., Belta, C., Kumar, V.: Controlling three dimensional swarms of robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 964–969 (2006)
22. Reif, J.H., Wang, H.: Social potential fields: A distributed behavioral control for autonomous robots. Robotics and Autonomous Systems **27**(3), 171–194 (1999)
23. Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. In: Computer Graphics, pp. 25–34 (1987)
24. Siegwart, R., Nourbakhsh, I.R.: Introduction to Autonomous Mobile Robots. Bradford Company, Scituate, MA, USA (2004)
25. Snape, J., van den Berg, J., Guy, S., Manocha, D.: The hybrid reciprocal velocity obstacle. IEEE Transactions on Robotics **27**(4), 696 –706 (2011)
26. Tan, K.H., Lewis, M.: Virtual structures for high-precision cooperative mobile robotic control. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 132–139 (1996)