Prioritized Group Navigation with Formation Velocity Obstacles

Ioannis Karamouzas and Stephen J. Guy

Abstract— **We introduce the problem of navigating a group of robots having prioritized formations amidst static and dynamic obstacles. Our formulation allows users to define a number of template formations, each with a specified priority value. At each planning cycle, we compute a new formation which accounts for both these priority values and the safe progress of the robots towards their goal. To this end, we introduce a new velocity-based navigation approach which we denote as Formation Velocity Obstacles (FVO). Like other velocity-based approaches, FVO allows anticipatory collision avoidance accounting for the likely future motion of nearby obstacles. However, we extend these previous approaches and allow anisotropic agents which rotate themselves to orient along their direction of travel. We integrate these FVOs with a Bayesian framework to infer priority values for arbitrary formations from the user-given templates. The result is a complete framework for prioritized formation planning.**

I. Introduction

Multiple robots walking together as a group while maintaining specific configurations are nowadays commonly used to perform critical tasks, such as search and rescue operations, exploration and security patrols. In robotics, the formation control problem has been extensively studied and different methods have been proposed to control the dynamics and stability of a group formation, including behavioral-based approaches [1], [2], leader-follower models [3], [4], virtual structure techniques [5], [6], social potential fields [7] and roadmap-based methods [8], [9]. Recently, the problem of pattern formation has also been studied, where the task is to generate smooth and collision-free transitions between arbitrary formations in obstacle-free environments [10], [11].

In contrast, in this work, we focus on the problem of safely navigating a group of robots amidst static and dynamic obstacles. Our approach allows the robots to deform or completely change their formation to navigate more efficiently in the presence of obstacles. To this end, we introduce the concept of *prioritized formations*, a list of user-defined template formations, each having a numeric priority value associated with it that indicates the user's preference of the formation for the navigation task at hand.

The use of prioritized formations is motivated by the navigational challenges that a group of robots has to face while performing critical tasks. Consider, for example, a team of robotic scouts sent to explore an area. Typically, we may require the robots to walk in a tactically valuable pattern, such as in a line formation, to gather as much information as possible. However, in highly constrained settings, like a narrow passage, the robots may have to employ a column formation so that they can still move as a coherent group while keep exploring the area. To this end, we propose a framework that dynamically chooses group formations by balancing the user-defined priorities with the constraints imposed by obstacles and other agents present in the environments. To do so, we address two distinct aspects of the prioritized formation navigation problem: evaluation of arbitrary formations and guaranteed collision-free motion.

In practice, a group of robots will rarely adopt one of the exact template formations due to obstacles, navigational uncertainty, and other local interactions. Likewise, as a group transitions from one formation to another, the priority value of the transient formation will be unspecified. Therefore, we propose a *Bayesian* interpolation framework to infer the values of these emergent formations from the user specified priorities of the template formations.

Additionally, steering a given group formation in a dynamic environment is a challenging task in and of itself. Standard local navigation techniques based on the notion of Velocity Obstacles [12], [13], provide a robust solution to the multi-robot navigation problem and have also been used to steer non-holonomic robots [14], [15], [16], maintain team coherence [17], navigate swarms [18], and simulate the local dynamics of pedestrian groups [19]. However, most of the VO-based techniques assume that the navigating robot can be well represented as a disc or, more recently, as a freely rotating rectangle [20]. These assumptions, though, do not hold for many formations, as they often have wide aspect ratios (e.g., several robots walking line-abreast) or have a single leader (e.g., an inversed V-like formation). In these cases, it is important that the formation rotates to maintain its specified orientation as its members move through the environment. We therefore introduce the concept of *formation velocity obstacles* (FVO) for guaranteed collision-avoidance with static and moving obstacles. Our formulation approximates a formation as an oriented bounding box while accounting for the non-holonomic nature of oriented travel. Our proposed FVO can be used to steer any anisotropic entity; group steering is one example of its applicability.

Overall, this work has three main contributions. The first is a mathematical formalization of the group navigation problem with prioritized formations. The second is a Bayesian approach for evaluating arbitrary formations. Third, is the introduction of the FVO formulation to account for the rotation of formations as they navigate through the environment.

The remainder of this paper is organized as follows. Section II formalizes our planning problem and provides an overview of our approach. Section III presents our Bayesian

The authors are with the Department of Computer Science and Engineering, University of Minnesota, USA. Email: {ioannis,sjguy}@cs.umn.edu

method for inferring priority values of formations and Section IV details the FVO formulation. Section V presents simulation results obtained with our framework and finally, some conclusions and plans for further research are discussed in Section VI.

II. Overview

A. Notation

Throughout this paper, we denote scalars x in lower case italics, vectors $\mathbf{x}$ in lower case bold, and sets of (positional or velocity) vectors X in upper case italics. Furthermore, we denote a normalized vector by $\widehat{\mathbf{x}}$, the perpendicular vector of $\mathbf{x}$ by $\mathbf{x}^{\perp}$, and a unit vector pointing in direction θ by $\boldsymbol{v}_{\theta} = [\cos\theta, \sin\theta]^T$. We notate an open line segment having $\mathbf{a}$ and $\mathbf{a}+\mathbf{b}$ endpoints as:

$$L(\mathbf{a}, \mathbf{b}) = \{\mathbf{a} + s\mathbf{b} \,|\, s \in (0, 1)\}, \tag{1}$$

an oriented bounding box of half-height h and half-width w centered at $\mathbf{p}$ and oriented towards $\boldsymbol{v}$ as:

$$B(\mathbf{p}, \boldsymbol{v}, h, w) = \{\mathbf{p} + a\boldsymbol{v} + b\boldsymbol{v}^{\perp} \mid |a| \leq h, |b| \leq w\}, \tag{2}$$

and the Minkowski sum of two sets as:

$$X \oplus Y = \{\mathbf{x} + \mathbf{y} \,|\, \mathbf{x} \in X, \mathbf{y} \in Y\}. \tag{3}$$

B. Problem Definition

We are given a group of n holonomic robots that have to move in formation through an environment. We assume that an expert user has provided a set of k template formations $T = \{T_1, \ldots, T_k\}$ and a priority value p_i indicating the desirability of each T_i, with higher priorities corresponding to more preferred formations. Each template T_i consists of n positions, one for each agent; we treat these templates as "loose" formations in that any agent can occupy any position (as long as each position is covered) and allow small deformations of the template if needed for navigation.

The problem, then, is characterized as follows. The robots need to reach a specified goal area while maintaining a configuration that is as close as possible to the template formations and avoiding collisions with each other and with the static and dynamic obstacles present in the environment. We assume any obstacle can be represented as an open circular disc. Non-circular obstacles are approximated either as a bounding disc or as a union of several discs. We further assume that the group members move on the 2D plane and are represented as discs.

C. Overall Approach

Following [19], we plan paths in an online fashion for each group member using a two-phase approach. In the first phase, we compute a new formation and collision-free velocity for the entire group of the robots and in the second phase, we determine a new collision-free velocity for each robot. Both phases follow the traditional sensing-acting cycle with time step Δt.

Group Planning Phase: In this phase, we seek to find, at each timestep, a formation F^* that balances the user's preferences and the collision-free progress of the group towards its goal. Given any arbitrary formation F, we use the following fitness function to account for such a balance:

$$E(F) = p_{F}\,(\mathbf{v}_{F} \cdot \hat{\mathbf{v}}^{\text{pref}}), \tag{4}$$

where p_F denotes the priority of F, $\mathbf{v}^{\text{pref}}$ is the preferred velocity of the group pointing towards the goal, and $\mathbf{v}_F$ is the optimal collision-free velocity that the group can obtain when adopting F. Consequently, a formation with $\mathbf{v}_F = \mathbf{0}$ will always have a lower fitness than any other formation that allows the group to safely progress more closer to its goal. Similarly, given two formations with the same optimal collision-free velocity, the one with the highest priority will always be preferred.

Given Eq. (4), we can then formulate the group planning phase as an optimization problem of selecting at each time step a new optimal formation F^* for the group. To find such a formation, we would ideally optimize over all possible formations that a group can adapt. However, in all of our experiments, it was sufficient to evaluate only the template formations and the current configuration of the robots. Strategies for evaluating a broader selection of formations are discussed in Section VI. Furthermore, to evaluate the fitness of any given formation, we should be able to infer its priority (if it is not a template formation) and compute its optimal collision-free velocity. We address these issues in Section III and Section IV, respectively.

Robot Planning Phase: In the second phase of our approach, we use the new optimal formation of the group and its corresponding collision-free velocity to plan the individual motions of the group members. In particular, F^* is oriented towards its new direction of travel and extrapolated into the near future, as in [19]. The extrapolated formation determines the new intermediate goals for the robots. This essentially becomes a pattern matching problem of the style proposed in [11], where n robots have to transition from their current to a new formation in a collision-free manner. Each robot is given a goal position in the new formation which minimizes the overall displacement of the robots. This goal then defines the new preferred velocity for the robot which serves as input to the robot's local navigation routine. In our framework, we use the ORCA navigation routine [13], which efficiently computes an optimal collision-free velocity that is as close as possible to the preferred one.

Because the group planning phase of our approach is the major novelty, we will focus on that phase in the rest of this paper. We refer to [11], [19] for more details regarding the robot planning phase.

III. Bayesian Formation Interpolation Scheme

To evaluate a given formation F, we must be able to infer its priority value from the set of template formations T. Within the context of the prioritized navigation, such inference must meet three important criteria. First, if a formation matches exactly one of the template formations, it should have the exact same priority as that template

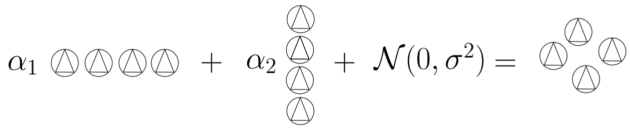


Fig. 1. **Defining Arbitrary Formations** We can decompose any arbitrary formation into a linear combination of the user provided template formations T plus some noise. For example, the staggered formation on the far right is a combination of the line-abreast and column formations with $a_1 = 0.56$ and $a_2 = 0.43$, respectively, and $\sigma = 0.08$.

formation. Second, no formation can have a higher priority than that of the top ranked template formation. Thus, in the absence of constraints, the robots will always follow the highest priority template. Third, formations that are very close to one of the template formations should receive a priority value very close to the priority of that template. This allows for small deformations in the group's formation when needed for navigation.

To satisfy these criteria, we formulate the problem of determining p_F as a Bayesian inference problem. In our framework, we use this Bayesian scheme to determine the weight of the current group formation. However, this approach can be applied to any arbitrary formation. Given a formation F, we assume it can be defined as a convex combination of the k provided templates in T, plus some (Gaussian) noise:

$$F = a_1 T_1 + \cdots + a_k T_k + \mathcal{N}(0, \sigma^2), \text{ s.t } \sum_{i=1}^{k} a_i = 1, \quad (5)$$

where $a_i \geq 0$, and σ denotes the standard deviation of the noise. We can interpret a_i as the proportion of the formation F that can be explained by the templates T_i. The size of σ determines how well F is captured by the weighted linear combination of the templates. An example of this decomposition is given in Fig. 1.

Given the weights a_i, we can infer the value of the formation's priority as follows:

$$p_F = a_1 p_1 + a_2 p_2 + \cdots + a_k p_k - \gamma\sigma, \quad (6)$$

where the term γ is used to weight how much deformation away from the provided templates is penalized. Evaluating Eq. (6) depends on inferring the set of values $a_1, \ldots, a_k$ and σ which are most likely given F:

$$\begin{aligned} &\operatorname*{argmax}_{a_1,\ldots,a_k,\sigma} P(a_1, \ldots, a_k, \sigma | F) \quad (7) \\ &= \operatorname*{argmax}_{a_1,\ldots,a_k,\sigma} \mathcal{L}(F | a_1, \ldots, a_k, \sigma) + \mathcal{L}(a_1, \ldots, a_k) + \mathcal{L}(\sigma), \end{aligned}$$

where $\mathcal{L}(\cdot)$ denotes the *log likelihood* function.

Assuming all template formations and all valid values for $a_1, \ldots a_k, \sigma$ are equally likely, then Eq. (7) is reduced to maximizing the first log likelihood term. We can model this term as the sum of the pairwise distances between the positions of the robots in formation F and their corresponding positions in the formation implied by the estimated parameters $a_1, \ldots, a_k$. Let $D(F,\, a_1 T_1 + \ldots + a_k T_k)$ be this distance. Following from the definition of a Gaussian distribution, we can therefore formulate our inference problem as:

$$\operatorname*{argmax}_{a_1,\ldots,a_k,\sigma} \frac{-[D(F,\, a_1 T_1 + \ldots + a_k T_k)]^2}{\sigma^2}. \quad (8)$$

Because the template formations do not specify an ordering of the robots, when computing D, we must take the additional step of finding the best matching between the positions of robots in F and the templates T_i.

We use the Expectation-Maximization algorithm to solve Eq. (8). After being initialized with a guess for values of $a_1, \ldots, a_k$ and σ, the algorithm proceeds in two steps. First, it computes $a_1, \ldots, a_k$ values that maximize the log-likelihood of the formation F (assuming a given value of σ). Then, these values are used to compute the most likely value for σ. Such value is computed directly from the standard deviation of the distance D. We keep alternating between the two steps until convergence.

IV. Oriented Formation Planning

Given a formation F, we need to be able to determine its best possible collision-free velocity $\mathbf{v}_F$ in a manner which accounts for the formation reorienting itself along its direction of travel. We assume that the group has a preferred speed and direction which we denote as $\mathbf{v}^{\text{pref}}$. Then, we define the optimal velocity as the velocity which is as close as possible to $\mathbf{v}^{\text{pref}}$ while still avoiding all upcoming collisions. We can combine this velocity $\mathbf{v}_F$ with the formation's weighting p_F (as computed in Section III) to evaluate its fitness using Eq. (4).

A. Rotationally-invariant Velocity Obstacles

Our approach to computing $\mathbf{v}_F$ builds on the concept of Velocity Obstacles (VO) introduced in [12]. More formally:

Definition 1. The *velocity obstacle* $VO^{\tau}_{A|O}$ between a robot A and a (moving) obstacle O is the set of all *relative* velocities of A with respect to O that will result in a collision between A and O before some time horizon τ.

Previous work has defined VOs for rotationally-invariant robots (e.g., discs) or robots which translate without rotating [12], [14], [13]. However, such a formulation is not particularly appealing to group formations as it leads to overly-approximated VOs. Hence, many collision-free velocities will be characterized as inadmissible and cannot be selected by the group, while in other cases, a VO cannot be even defined (see Fig. 2). Furthermore, in a traditional, rotationally-invariant VO, an obstacle's velocity can be accounted for by moving the apex of the VO to lie at the obstacle's velocity. However, such a strategy cannot be employed for an anisotropic formation because the *absolute* velocity determines the orientation of the formation (not the relative one). To address these issues, we define below the formation velocity obstacles.

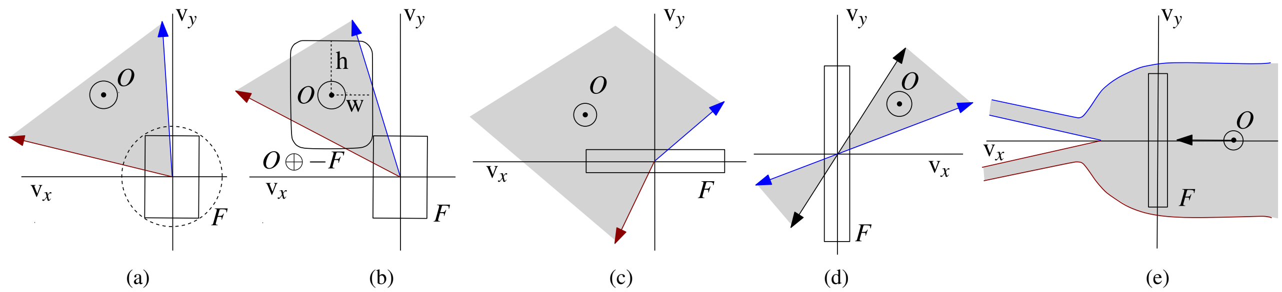


Fig. 2. **Collision Avoidance Using Velocity Obstacles.** (a) A rotationally-invariant VO that approximates F as disc to avoid a static obstacle O. (b) The FVO for the same configuration allows more flexibility in the velocities that a formation can take leading to more efficient motion. (c, d) The obstacle is very close to the formation and no traditional VO could be defined. However, a well-defined FVO exists in these scenarios. (e) FVOs induced by dynamic obstacles are more complex, having non-linear boundaries.

B. Formation Velocity Obstacles

We derive a new VO formulation that explicitly accounts for the rotation of a formation as it *aligns* to its direction of motion. Rather than plan for the exact formation, we consider the bounding box that encloses all robots in the formation. For ease of notation, throughout the rest of this section, we will refer to this box as F. Let w_F and h_F denote the half-width and half-height, respectively, of F where the width is defined along the axis perpendicular to the direction of travel. Without loss of generality, we assume that F is centered at the origin and oriented along the positive y-axis. Formally, $F = B(\mathbf{0}, [0, 1]^T, h_F, w_F)$ as defined in Eq. (2).

Then, for a given orientation $\boldsymbol{v}_\theta$, the formation velocity obstacle is defined as follows:

Definition 2. The *formation velocity obstacle* $FVO^{\theta,\tau}_{F|O}$ (read: the formation velocity obstacle for F induced by O for time horizon τ and orientation θ) is the set of all velocities of F that will result in a collision between F and O before time τ, assuming F instantaneously rotates to lie along $\boldsymbol{v}_\theta$.

More formally, let $\mathbf{p}_O$ be the position of the obstacle O having radius r_O. Let also $w = w_F + r_O$ and $h = h_F + r_O$ be the half-width and half-height, respectively, of the oriented bounding box enclosing the Minkowski sum $O \oplus -F$. Then, we can conservatively approximate the FVO by performing an OBB-point swept test (see Fig. 3b). For a given orientation $\boldsymbol{v}_\theta$, a velocity $\mathbf{v} = \kappa \boldsymbol{v}_\theta$, $\kappa \in \mathbb{R}$ leads to a collision at some time t only if the following two conditions are met:

$$|(\mathbf{p}_O + t\mathbf{v}_O) \cdot \boldsymbol{v}_\theta^\perp| < w \tag{9}$$

$$|(\mathbf{p}_O + t\mathbf{v}_O - t\mathbf{v}) \cdot \boldsymbol{v}_\theta| < h \tag{10}$$

From Eq. (9) the following lower t^- and upper bounds t^+ on t can be defined:

$$t^\pm = \frac{\pm w - \mathbf{p}_O \cdot \boldsymbol{v}_\theta^\perp}{\mathbf{v}_O \cdot \boldsymbol{v}_\theta^\perp}. \tag{11}$$

Dividing Eq. (10) by t and rearranging gives:

$$\left(\frac{\mathbf{p}_O \cdot \boldsymbol{v}_\theta - h}{t} + m\right)\boldsymbol{v}_\theta < \mathbf{v} < \left(\frac{\mathbf{p}_O \cdot \boldsymbol{v}_\theta + h}{t} + m\right)\boldsymbol{v}_\theta, \tag{12}$$

where $m = \mathbf{v}_O \cdot \boldsymbol{v}_\theta$. Equation 12 defines all velocities $\mathbf{v}$ that will lead to a collision between F and O for a given θ.

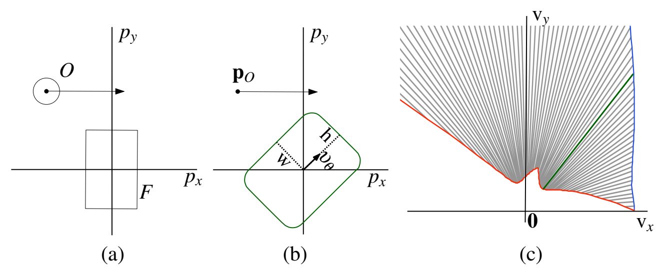


Fig. 3. **Dynamic Obstacles.** (a) An obstacle O moves in front of the formation F. (b) For a given orientation $\boldsymbol{v}_\theta$, the set of velocities that lead to a collision between F and O lies along a line segment. (c) The resulting FVO can be found as the union of all line segments. Here, we sample the segments using a separation in orientation of 1.8°. The green segment corresponds to the orientation shown in (b).

Hence, the formation velocity obstacle for all orientations $\theta \in [-\pi, \pi]$ is a collection of line segments:

$$FVO^\tau_{F|O} = \bigcup_{\substack{\theta \in [-\pi,\pi] \\ t \in (0,\tau) \cap (t^-, t^+)}} L\left(\left(\frac{\mathbf{p}_O \cdot \boldsymbol{v}_\theta - h}{t} + m\right)\boldsymbol{v}_\theta, \frac{2h\boldsymbol{v}_\theta}{t}\right). \tag{13}$$

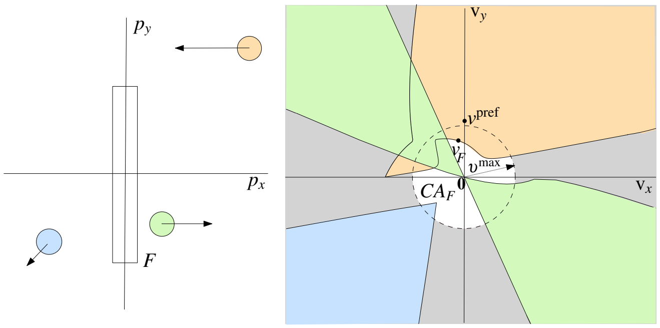
This equation defines a full FVO valid over all orientations θ. Formally:

Definition 3. The *formation velocity obstacle* $FVO^\tau_{F|O}$ for F induced by an obstacle O is a union of line segments as defined in Eq. (13).

By connecting the end points of the line segments, an explicit representation of the boundary of the formation velocity obstacle can be obtained. We refer the reader to Fig. 3 for an example. We can similarly compute the boundary of the formation velocity obstacle when O is static by setting $\mathbf{v}_O = \mathbf{0}$ in Eq. (13). However, in this case, a closed form expression can also be obtained for the boundary, as discussed in the Appendix and shown in Fig. 2.

C. Choosing a Collision-free Velocity

Having computed the $FVO^\tau_{F|O}$ for each static and dynamic obstacle present in the environment, we need to determine an optimal new velocity $\mathbf{v}_F$ for the formation F that lies outside the union of FVOs. We also assume that the group members and, subsequently, F, are subject to a maximum speed constraint $v^{\max}$ modeled as a disc



(a) (b)

Fig. 4. **Multiple Obstacles.** (a) A formation F navigating amidst multiple moving obstacles. The arrows indicate current velocities. (b) The FVOs induced by the obstacles for infinite time horizon. The white region denotes the set of safe new velocities for F, and $\mathbf{v}_F$ is the optimal new velocity.

$C(\mathbf{0}, v^{\max})$ centered at the origin of the velocity space). Consequently the set of feasible velocities CA_F that F can choose (Fig. 4) is :

$$CA_F = C(\mathbf{0}, v^{\max}) \setminus \bigcup_{O} FVO^{\tau}_{F|O}. \tag{14}$$

Additional constraints on velocity or acceleration can be similarly incorporated to the FVO formulation in a similar manner or applying after-the-fact to the derived velocities.

Given the set CA_F, the optimal formation velocity $\mathbf{v}_F$ is, then, the allowed velocity closest to the preferred one $\mathbf{v}^{\text{pref}}$:

$$\mathbf{v}_F = \operatorname*{arg\,min}_{\mathbf{v} \in CA_F} \|\mathbf{v} - \mathbf{v}^{\text{pref}}\|. \tag{15}$$

Solving Eq. (15) is a difficult non-convex optimization problem. However, an approximate solution can be efficiently computed using 2D linear programming following the approach presented in the ORCA navigation framework [13]. Here, each VO is approximated by a single linear constraint that lies tangent to the VO at the point closest to the preferred velocity. Assuming all VOs are convex, the ORCA approach provides a conservative approximation which still guarantees to avoid all collisions while running in linear complexity in the number of VO constraints. In our case, since the FVOs are typically non-convex (e.g., Fig. 3), we consider the orthogonal projection of the preferred velocity on the boundary of each FVO for the ORCA linearization [21]. The $\mathbf{v}_F$ computed in Eq. (15) is combined with Eq. (4) to determine the target formation for the robots having the highest overall score. Since we assume that F can instantaneously rotate, there may be collisions as the robots transition between their current formation to the new one. Such collisions are handled individually by the robots using ORCA as explained in the *Robot Planning Phase* of Section II-C.

V. Experiments and Results

We implemented our framework in C++ and ran several simulated experiments over a variety of scenarios spanning several different types of formations, different group sizes, and different types of environments. During the EM optimization of Eq. (8) we used 500 steps of a stochastic gradient descent approach based on simulated annealing and 2 EM loops. For determining the FVOs, as specified by Eq. (13), we discretized the boundaries in 200 segments of 1.8° and set the time horizon to 5 s. All simulations ran in realtime (over 100 fps) on an Intel 2.4 GHz Core 2 Duo processor (on a single thread). This section highlights several key example scenarios and performance results. Simulation videos are available at `http://motion.cs.umn.edu/r/FVO/`.

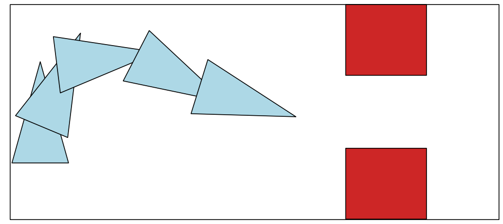


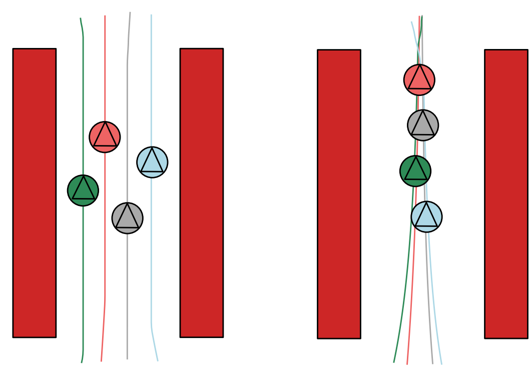
Fig. 5. **Formation Rotation.** A formation modeled as an oriented, isosceles, triangle has to navigate through a narrow passage. Only by rotating in the direction of travel (as allowed by FVO) can the formation fit through. Here the maximum angular speed is capped to $\pi/3$ rad/sec.

A. *Formation Rotation*

By using our newly proposed FVOs, we can allow formations to reorient themselves to be aligned to their direction of travel. The advantage of this approach can be clearly seen in Fig. 5. Here, a triangle-shaped formation has to move through a narrow passage. Since no other template formation is given, if the triangle does not rotate, it will not fit through. The same applies if a disc rather than a bounding box is used to compute the velocity obstacles induced by the environment.

B. *Dynamic Formation Weighting*

The Bayesian formation weighting scheme allows us to infer values for arbitrary formations given a small number of user-weighted template formations. The user-selected parameter γ has an important effect on how arbitrary formations are weighted. The effect of this parameter can be seen in the *Squad* scenario shown in Fig. 6. Here a group of four agents must navigate past various buildings while maintaining squad formations such as walking in a line-abreast or a column



(a) (b)

Fig. 6. **Effect of γ parameter.** Agents navigate through a passage given two formations: line abreast and single column. (a) With a small value of γ agents adopt an ad-hoc formation which fits the obstacle. (b) With a larger value of γ agents follow very closely the single column formation.

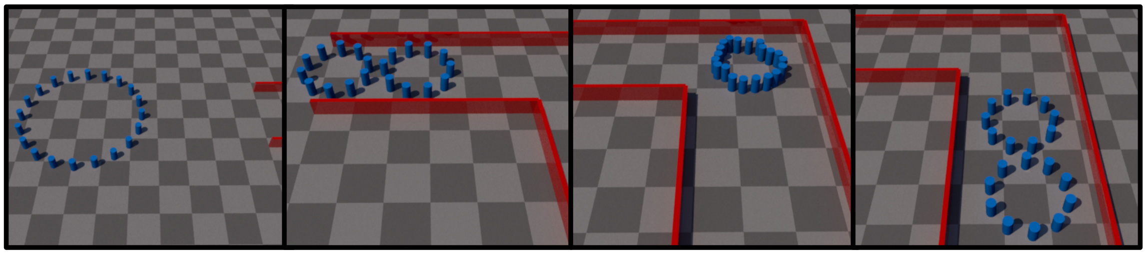
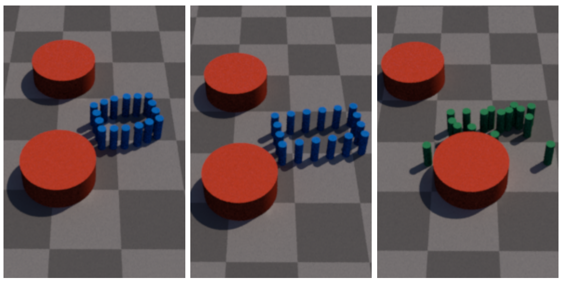


Fig. 8. **Circle Scenario.** A group of twenty agents navigates through a static environment while maintaining one of a three user-defined template formations with various priorities: a large circle (p=9), two smaller circle (p=6), and a single small circle (p=1). By using our approach, agents automatically choose the highest priority formations that avoid collisions, smoothly transition between formations, and can dynamically reorient and deform the formations as needed for navigation.



(a) (b) (c)

Fig. 7. **Dynamic Obstacle Comparison.** Our approach allows for navigation of formations amongst dynamic obstacles. (a) With fast moving obstacles the formation has to deform to avoid collisions. (b) Little deformation is needed with slower obstacles. (c) Even with slower obstacles, purely distributed planning methods such as ORCA fail to maintain formation.

formation. As the robots approach the narrow passage they must abandon the highly weighted line-abreast formation to fit. With a large value for γ the robots collapse down to the single column formation, with a small value the robots maintain an ad-hoc staggered formation that is not near to either of the input templates.

C. Dynamic Obstacles

Our approach is able to account for the motion of dynamic obstacles in the environment. In the scenario shown in Fig. 7, agents are given the goal of walking in a box formation past a series of moving discs. When the disc obstacles are moving quickly (Fig. 7a), the agents must deform their box shape in order to maintain a collision-free trajectory. When the discs are moving more slowly (Fig. 7b), the agents can avoid collision without deformations. When ORCA is used in the same scenarios, the agents successfully avoid collisions, but fail to maintain a cohesive formation (Fig. 7c).

D. Navigation with Roadmaps

Our approach extends to simulations of multiple agents navigating with complex formations in multi-obstacle environments where roadmaps may be needed. An example of roadmap-based navigation is shown in the *Circle* scenario depicted in Fig. 8, where a group of 20 agents follows a roadmap to move through a bent corridor in a variety of circularly shaped formations. In this scenario, the top priority formation is a large circle, which is too big to fit in the hallway. While the large circle deforms at first, the agents soon adopt the second ranked formation of a double circle which can fit in the hallway undeformed. However, this formation is still too wide to turn the corner, so agents must then adapt their configuration to the least preferred formation of a tightly packed circle. After making the turn, the group reorients itself to walk down the rest of the corridor in the double circle formation.

TABLE I

PERFORMANCE ANALYSIS ON DIFFERENT SCENARIOS

Scenario	# Agents	# Obstacles	Total time (s)	Time per frame (ms)
Squad	4	12	0.078	0.394
Dynamic Obstacles	16	40	0.335	0.914
Circle	20	60	1.248	1.259

E. Performance

In all scenarios, agents navigated successfully to their goals while avoiding collisions with each other and the environment. The path for all robots were computed in realtime (see Table I). This is consistent with the polynomial theoretical runtime expectations of our approach. The constrained optimization problems generated by FVO can be solved in O(kn) randomized expected time, where n is the number of constraints (static + dynamic obstacles), and k the number of formations that the group evaluates. For the Bayesian formation weighting, every formation evaluation starts with determining the best assignment between the current agent positions and each of the template formations. Each of these matching problems can be solved in polynomial time O(n^3) [22], where n is the number of group members. Since, though, Eq. (8) is only computed once per time step, it does not affect the real time performance of our approach.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have introduced the concept of prioritized formations for guiding a group of robots amidst static and dynamic obstacles. Our approach plans over a joint formation-velocity space employing a Bayesian scheme for evaluating the priority values of formations and the notion of *formation velocity obstacle* that accounts for the anisotropic nature of rotating formations. It produces convincing collision-free motions for the group and runs in real time.

In the future, we would like to address several limitations of our approach. Currently, we conservatively approximate the space that a formation occupies with a bounding box. While this approximation is sufficient for many scenarios, a more complex representation that considers the convex hull of the robots can potentially generate a further range of motions, particularly in highly dense settings. Additionally, in our current implementation, a group does not sample broadly over all possible formations that it can adopt. More formations to evaluate can be obtained by blending between the template formations or sampling over a representative formation derived from the templates.

Our current results have been presented on simulated agents; to implement our approach on a team of physical robots, a leader robot must be chosen to compute the high-level formation plan. Thus, in the future, we would like to develop a more fully decentralized approach. Other interesting areas for further development include allowing the automatic splitting and merging of a group into multiple formations [23], as well as combining our prioritized local planner with a global planner that considers the feasibility of group formations [24]. We would also like to model interactions between multiple groups by deriving an FVO-based formulation for reciprocal collision avoidance. Another potential application is to use our velocity obstacle formulation for non-holonomic robots, such as cars. This will require accounting for kinodynamic constraints as well as further extending our formulation.

REFERENCES

[1] T. Balch and R. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[2] J. Lawton, R. Beard, and B. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 6, pp. 933–941, 2003.

[3] J. Desai, J. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, p. 905, 2001.

[4] T. D. Barfoot and C. M. Clark, "Motion planning for formations of mobile robots," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 65–78, 2004.

[5] M. Lewis and K. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, 1997.

[6] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *IEEE Conf. on Decision and Control*, 2001, pp. 2968–2973.

[7] T. Balch and M. Hybinette, "Social potentials for scalable multi-robot formations," in *IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 73–80.

[8] O. Bayazit, J.-M. Lien, and N. Amato, "Better group behaviors in complex environments using global roadmaps," in *Int. Conf. on Artificial life*, 2003, pp. 362–370.

[9] A. Kamphuis and M. H. Overmars, "Motion planning for coherent groups of entities," in *IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 3815–3822.

[10] S. Takahashi, K. Yoshida, T. Kwon, K. Lee, J. Lee, and S. Shin, "Spectral-based group formation control," *Computer Graphics Forum*, vol. 28, no. 2, pp. 639–648, 2009.

[11] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley, "Image and animation display with multiple mobile robots," *Int. J. of Robotics Research*, vol. 31, no. 6, pp. 753–773, 2012.

[12] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. of Robotics Research*, vol. 17, pp. 760–772, 1998.

[13] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Int. Symp. on Robotics Research*, ser. Springer Tracts in Advanced Robotics, vol. 70, 2011, pp. 3–19.

[14] D. Wilkie, J. van den Berg, and D. Manocha, "Generalized Velocity Obstacles," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 5573–5578.

[15] J. van den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 3475–3482.

[16] M. Rufli, J. Alonso-Mora, and R. Siegwart, "Reciprocal collision avoidance with motion continuity constraints," *IEEE Trans. on Robotics*, vol. 29, no. 4, pp. 899–912, 2013.

[17] A. Kimmel, A. Dobson, and K. Bekris, "Maintaining team coherence under the velocity obstacle framework," in *Int. Conf. on Autonomous Agents and Multiagent Systems - Volume 1*, 2012, pp. 247–256.

[18] V. G. Santos, M. F. Campos, and L. Chaimowicz, "On segregative behaviors using flocking and velocity obstacles," in *Distributed Autonomous Robotic Systems*, ser. Springer Tracts in Advanced Robotics, 2014, vol. 104, pp. 121–133.

[19] I. Karamouzas and M. Overmars, "Simulating and evaluating the local behavior of small pedestrian groups," *IEEE Trans. on Visualization and Computer Graphics*, vol. 18, no. 3, pp. 394–406, 2012.

[20] A. Giese, D. Latypov, and N. M. Amato, "Reciprocally-rotating velocity obstacles," in *IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 3234–3241.

[21] S. Guy, J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey, "Clearpath: highly parallel collision avoidance for multi-agent simulation," in *ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, 2009, pp. 177–187.

[22] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. of the ACM*, vol. 19, no. 2, pp. 248–264, 1972.

[23] T. Huang, M. Kapadia, N. I. Badler, and M. Kallmann, "Path planning for coherent and persistent groups," in *IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 1652–1659.

[24] A. Krontiris, S. Louis, and K. E. Bekris, "Multi-level formation roadmaps for collision-free dynamic shape changes with nonholonomic teams," in *IEEE Int. Conf. on Robotics and Automation*, 2012, pp. 1570–1575.

APPENDIX

A closed form solution exists for determining the $FVO^{\tau}_{F|O}$ boundaries for a formation F induced by a static obstacle O positioned at $\mathbf{p}_O$. Let w, h denote the half-width and half-height, respectively, corresponding to the (oriented) bounding box that encloses the Minkowski sum $O \oplus -F$ (see Fig. 2b). Let us also consider a ray $\lambda = s\boldsymbol{v}, s \geq 0$, starting at the origin and heading in the normalized direction $\boldsymbol{v}$. The problem can then be formulated as finding the direction $\boldsymbol{v}$ such that the heading of the oriented bounding box is aligned with $\boldsymbol{v}$. This leads to the following solution:

$$\begin{aligned} \boldsymbol{v}^{\pm} &= \begin{bmatrix} \ell & \pm w \\ \mp w & \ell \end{bmatrix} \frac{\mathbf{p}_O}{\|\mathbf{p}_O\|^2} \\ s^{\pm} &= h \pm \ell, \end{aligned} \tag{16}$$

where $\ell = \sqrt{\|\mathbf{p}_O\|^2 - w^2}$. The positive solution leads to the right boundary ($\boldsymbol{v}^+$) of the FVO; using the negative solution, the direction $\boldsymbol{v}^-$ of the left boundary is obtained. The resulting FVO is a truncated cone with its apex at the origin and its sides tangent to the disc of radius w centered at $\mathbf{p}_O$. See Fig. 2b.

The same approach can be employed to compute the FVO when the distance between the obstacle and the formation is less than w as in Fig. 2c. It can be easily shown that the boundaries of the resulting FVO are perpendicular to the two tangents from the origin to the disc centered at $\mathbf{p}_O$ having a radius of h.