Goal Velocity Obstacles for Spatial Navigation of Multiple Virtual Agents (Extended Abstract)

Jamie Snape
snape@cs.unc.edu

Dinesh Manocha
dm@cs.unc.edu

Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599

ABSTRACT

We present the *goal velocity obstacle* for the spatial navigation of multiple virtual agents to planar goal regions in the two-dimensional workspace. Our approach uses velocity obstacles not only to compute collision-avoiding velocities with respect to other agents, but also to specify velocities that will direct an agent toward its spatial goal region. We demonstrate shorter path lengths and fewer collisions with only microseconds of additional computation per agent per time step than velocity-based methods that optimize on a single, preferred velocity toward the goal of each agent.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems, intelligent agents, coherence and coordination*; I.2.9 [**Artificial Intelligence**]: Robotics; I.6.8 [**Simulation and Modeling**]: Types of Simulation—*discrete event, gaming, visual*

Keywords

Multiagent planning, motion planning, robot coordination, collective behaviors, implicit cooperation

1. INTRODUCTION

The spatial navigation of groups of multiple virtual agents to specified goal locations is an important problem in many video games, mobile robotics, and simulated environments. Large numbers of agents may be incorporated into game levels and simulated environments, and, often, they must interact with agents controlled by a player. In addition, a human-controlled agent may also represent the goal location of a group of pursuing agents. In this case, the group of agents must be able to converge on a possibly moving, planar goal region that is the footprint of the human-controlled agent.

In previous work [3,6,8], each agent, such as an autonomous robot or a virtual agent chooses an avoiding new velocity based on some optimization to make progress toward its goal. Commonly, this optimization is on a *preferred velocity* that is directed to a roadmap node [1] or a fixed point in the center of a navigation mesh edge or face [7]. However, these points often approximate planar goal regions, and this contraction of the goal region to a point can cause artifacts, such as collisions when several agents converge on a single point. Behavior would be improved if the agent could navigate to any point in a goal region. When goal regions are moving, optimizing on a preferred velocity ignores that the position of the goal region may have changed significantly by the time the agent has computed a new velocity. Hence, the trajectory of the agent will not necessarily be directed toward its goal region, and the lengths of paths to the goal region will be increased. If the velocity of the goal region were considered during the optimization of the velocity, then the motion of the agent toward its goal region would again be improved

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

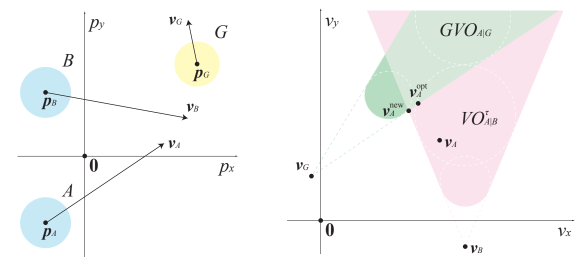

Figure 1: (left) Agent A navigating toward a moving goal region G with velocity $\boldsymbol{v}_G$ while avoiding agent B. (right) The *goal velocity obstacle* $GVO_{A|G}$ for agent A toward goal region G and the velocity obstacle $VO^\tau_{A|B}$ for agent A induced by agent B.

We introduce the *goal velocity obstacle* for navigating multiple agents to planar, spatial goal regions that counters the disadvantages of formulations that optimize on preferred velocity. The basic idea is that instead of only using velocity obstacles [3] to compute collision-avoiding velocities, we use them to define the goal regions of an agent within the velocity space. We call the velocity obstacle of an agent induced by its goal region a *goal velocity obstacle*, and if the agent chooses a velocity that is inside the goal velocity obstacle at each time step, then it will eventually reach its goal region. The goal velocity obstacle provides a unified formulation that allows for static or moving goals specified as points, line segments, and bounded, planar regions in two dimensions.

2. RELATED WORK

Video games and simulated environments have, historically, used force-based methods, such as flocking [5], in combination with roadmap [1] and navigation mesh [7] approaches to provide local collision avoidance for groups of

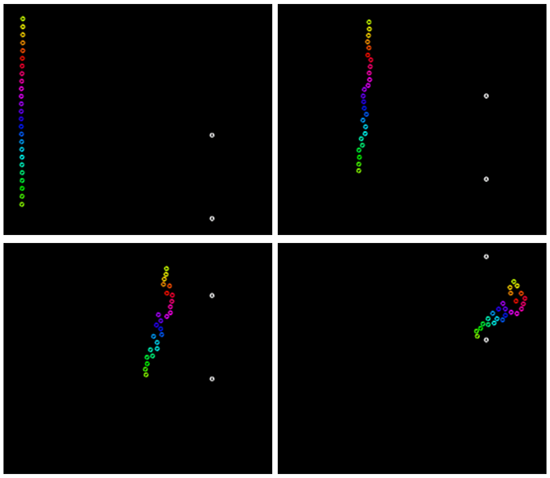

Figure 2: An experiment containing 25 agents navigating toward a moving goal region using goal velocity obstacles.

agents moving through the environment. Velocity-based methods, such as the velocity obstacle [3], and its derivatives, popular in mobile robotics, have exhibited improvements in terms of computational performance and local collision avoidance. Generally, current collision avoidance approaches are limited to using some form of point goal [8] or line segment goal [2] in connection with the global planner.

3. GOAL VELOCITY OBSTACLES

Instead of using velocity obstacles [3] purely for excluding velocities that may cause collisions with other agents or moving obstacles, then optimizing with respect to a preferred velocity for navigation to a goal in the workspace, we propose the additional use of velocity obstacles to define the goal regions of an agent within the velocity space. More precisely, we define the *goal velocity obstacle* of agent A toward the goal region G, denoted $GVO_{A|G}$, as

$$GVO_{A|G} = VO^{\tau}_{A|G} = \{\boldsymbol{v} \mid \exists s \in [0, \tau] :: s\,(\boldsymbol{v} - \boldsymbol{v}_B) \in G \oplus -A\}\,.$$

We then choose a new velocity $\boldsymbol{v}_A^{\text{new}}$ of agent A such that $\boldsymbol{v}_A^{\text{new}}$ lies not only *outside* the velocity obstacles induced by other agents, but also *inside* the goal velocity obstacle toward the goal region G, i.e., $\boldsymbol{v}_A^{\text{new}} \in GVO_{A|G} \setminus \bigcup_{A \neq B} VO^{\tau}_{A|B}$. Figure 1 shows the goal velocity obstacle of an agent toward a moving, disc-shaped goal region.

In general, there will be a choice of collision-free velocities $\boldsymbol{v}_A^{\text{new}}$ that will navigate the agent A to some point in its goal region. Assuming that there is no preference as to which point in the goal region an agent A ultimately reaches, we choose a velocity $\boldsymbol{v}_A^{\text{opt}}$, the *optimization velocity*, with respect to which we must optimize from those velocities that are collision-free and inside the goal velocity obstacle, i.e.,

$$\boldsymbol{v}_A^{\text{new}} = \arg\min_{\boldsymbol{v} \in GVO_{A|G} \setminus \bigcup_{A \neq B} VO^{\tau}_{A|B}} \left\| \boldsymbol{v} - \boldsymbol{v}_A^{\text{opt}} \right\|_2 .$$

Motivated by a desire for agents to make as minimal change in velocity as possible at each time step [4], we choose the optimization velocity $\boldsymbol{v}_A^{\text{opt}}$ of an agent A as follows. If the current velocity $\boldsymbol{v}_A$ is *inside* the goal velocity obstacle $GVO_{A|G}$, we choose the current velocity as the optimization velocity, whether or not that velocity is collision-free, i.e., $\boldsymbol{v}_A^{\text{opt}} = \boldsymbol{v}_A$. If the current velocity is *outside* the goal velocity obstacle, so the agent is moving away from its goal region, we choose the closest velocity to the current velocity $\boldsymbol{v}_A$ that lies inside the goal velocity obstacle, i.e.,

$$\boldsymbol{v}_A^{\text{opt}} = \arg\min_{\boldsymbol{v} \in GVO_{A|G}} \|\boldsymbol{v} - \boldsymbol{v}_A\|_2 .$$

The optimization velocity $\boldsymbol{v}_A^{\text{opt}}$ is distinct from the notion of preferred velocity $\boldsymbol{v}_A^{\text{pref}}$, and, in general, much less influences the path taken by the agent A.

4. CONCLUSION

We have presented the *goal velocity obstacle* for the spatial navigation of multiple agents to arbitrary-shaped, planar goal regions in the two-dimensional plane. Our approach uses velocity obstacles not only to compute velocities that may cause collisions with other agents, but also to define the goal velocity obstacle, which specifies velocities in the two-dimensional velocity space that will direct an agent toward its planar goal region.

We have applied our approach to multiple challenging experiments by integrating with the hybrid reciprocal velocity obstacle formulation [6] for collision avoidance. On average, the agents traverse shorter path lengths and have fewer collisions than when simply using preferred velocities directed to a single point in their goal region instead of goal velocity obstacles, adding only a few microseconds of computation time per agent at each time step. Screenshots of an experiment containing 25 agents navigating toward a moving goal are shown in Figure 2.

A full discussion of the goal velocity obstacle formulation, including a video and the results of our experiments, may be found online at `http://gamma.cs.unc.edu/GVO/`.

Acknowledgments

This work was supported in part by ARO Contract W911NF-04-1-0088, NSF Awards 1000579 and 1117127, and Intel.

5. REFERENCES

[1] J. Canny. *The Complexity of Robot Motion Planning.* MIT Press, 1988.

[2] S. Curtis, J. Snape, and D. Manocha. Way portals: efficient multi-agent navigation with line-segment goals. In *ACM SIGGRAPH Symp. Interact. 3D Graph. Games*, pages 15–22, 2012.

[3] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *Int. J. Robot. Res.*, 17:760–772, 1998.

[4] S. Guy, S. Curtis, M. Lin, and D. Manocha. Least-effort trajectories lead to emergent crowd behaviors. *Phys. Rev. E*, 85, 016110, 2012.

[5] C. Reynolds. Flocks, herds and schools: a distributed behavioral model. *ACM SIGGRAPH Comput. Graph.*, 21:25–34, 1987.

[6] J. Snape, J. v.d. Berg, S. Guy, and D. Manocha. The hybrid reciprocal velocity obstacle. *IEEE Trans. Robot.*, 27:696–706, 2011.

[7] G. Snook. Simplified 3D movement and pathfinding using navigation meshes. In *Game Programming Gems*, pages 288–304. Charles River, 2000.

[8] J. v.d. Berg, S. Patil, J. Sewall, D. Manocha, and M. Lin. Interactive navigation of multiple agents in crowded environments. In *ACM SIGGRAPH Symp. Interact. 3D Graph. Games*, pages 139–147, 2008.