

# Cooperative Pursuit with Voronoi Partitions

Haomiao Huang, Zhengyuan Zhou, Wei Zhang, Jerry Ding, Dušan M. Stipanović, and Claire J. Tomlin

**Abstract**—Consider a pursuit-evasion game where a number of pursuers are attempting to capture a single evader. Cooperation among multiple agents can be difficult to achieve, as this may require considering actions in the joint input space of all agents. This work presents a decentralized, real-time algorithm for cooperative pursuit of a single evader by multiple pursuers in bounded, simply-connected planar domains. The algorithm is based on minimizing the area of the generalized Voronoi partition of the evader. The pursuers share state information but compute their inputs independently. No assumptions are made about the evader’s control strategies other than requiring evader control inputs to conform to a limit on speed, and proof of guaranteed capture is shown when the domain is convex and the players’ models are identical. Simulation results are presented showing the effectiveness of this strategy, and experimental results using the pursuit strategy to guide human players in a pursuit-evasion game are also presented.

## I. INTRODUCTION

In this work we address a multi-agent pursuit-evasion game, with a number of pursuers attempting to capture a single evader in a simply connected planar region. The pursuers may have speed equal to or faster than the evader, and the objective is to find a successful cooperative strategy for the pursuers. Finding cooperation strategies among multiple agents in adversarial games can be difficult, as computing solutions over the joint state space of multiple agents can greatly increase computational complexity.

The challenge is to find a solution strategy that induces effective cooperation among the pursuers without incurring a significant computational burden. To accomplish this, we take an approach that allows computations to take place in the low dimensional configuration space of individual agents instead of the high dimensional joint state space of all agents. This allows solutions to be computed quickly and in real-time, resulting in a scalable pursuit algorithm that is both intuitive and easily implementable, while providing deterministic guarantees of capture under certain assumptions on agent dynamics and environment geometry.

In particular, we present a decentralized pursuit strategy based upon the Voronoi decomposition of the game domain with respect to agent positions. We prove that, under the assumptions of convex environments, kinematic agent dynamics, and equal maximum speeds, this strategy results in guaranteed

This work has been supported in part by NSF under CPS:ActionWebs (CNS-931843), by ONR under the HUNT (N0014-08-0696) and SMARTS (N00014-09-1-1051) MURIs and by grant N00014-12-1-0609, by AFOSR under the CHASE MURI (FA9550-10-1-0567).

H. Huang, Z. Zhou, J. Ding, and C. J. Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA {haomiao, zhengyuan, jding, tomlin}@eecs.berkeley.edu

W. Zhang is with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH, USA zhang@ece.osu.edu

D. M. Stipanović is with the Control and Decision Group, Coordinated Science Laboratory, University of Illinois, Urbana, IL, USA dusan@illinois.edu

capture of the evader in finite time. The pursuers cooperatively minimize the area of the evader’s Voronoi partition, which represents the set of all points that the evader can move to without being captured along the way by a pursuer. Each pursuer influences the evader’s partition only through the shared Voronoi boundary. Thus, each pursuer’s input decouples from that of the other pursuers and can be computed independently. However their inputs are coupled through the evader’s partition, giving rise to cooperation between the pursuers. The pursuit algorithm is decentralized in the sense that the pursuers compute their control actions independently given the agent positions, which is the only shared information. Furthermore, due to the analytic representation of the Voronoi partition in convex domains, the control inputs for the pursuers can be computed analytically, allowing for real-time implementation.

While the convex and equal-speed case may be limited in a practical sense, it provides useful insights into the selection of pursuit strategies for more complex multi-agent pursuit-evasion scenarios. Towards this goal, we generalize the Voronoi pursuit strategy to non-convex game domains and unequal agent speeds, and show how to apply a modified fast marching method (FMM) [1] to quickly compute generalized Voronoi partitions and pursuer inputs. Although theoretical guarantees are not yet found for capture in the more general scenario, this approach provides a scalable, computationally efficient, and easily implementable algorithm for computing cooperative pursuer inputs in a multi-pursuer scenario.

Some elements of the results for convex domains and equal speeds were first presented in [2]. This work elaborates upon the previous results, extends the algorithm to more general domain geometries and agent speeds, and includes new simulation and experimental results. Through the simulation results, we demonstrate the computational speed of the pursuit strategy, as well as the effect of Voronoi partitions on agent cooperation. In particular, it is shown that the algorithm encourages effective cooperative pursuit among multiple agents, resulting in superior performance to techniques such as the pure-pursuit, in which the pursuers attempt to minimize the instantaneous distance to the evader. This ability to encourage cooperation is one of the key advantages of the strategy proposed in this work.

In addition, we also present experimental results for games involving human agents, to demonstrate the use of our proposed approach to support human decision making. Specifically, the Voronoi partitions of the evader is used to create intuitive visualizations for a pursuing human agent. In experiments based on the BErkeley Autonomy and Robotics in CApture-the-flag Testbed (BEARCAT), human pursuers utilizing the pursuit strategy with appropriate visualization of the set were able to capture an evader even in the presence of uncertainties such as GPS noise and communication delays.

The paper is organized as follows. First, we discuss the relation of our work to previously proposed approaches in

Section II. Then, we provide a precise definition of the cooperative pursuit problem in Section III. Section IV describes the Voronoi pursuit strategy for agents with equal speeds in convex domains, along with a formal proof of guaranteed capture. With the intuition arising from this scenario, we extend the pursuit strategy to unequal speeds and non-convex simply connected domains in Section V, through the use of FMM to compute generalized Voronoi partitions. To demonstrate the value of our approach, a number of simulations are shown in Section VI comparing the performance of the proposed strategy against the pure-distance pursuit strategy, while experimental results are also presented in Section VII for games involving human agents. Finally, Section VIII concludes the paper with a discussion of the algorithms and results.

## II. RELATED WORK

Within a general theoretical setting, the class of pursuit-evasion games considered in this paper can be treated as a multi-agent differential game. The solution to such a problem can be obtained, at least in principle, from an appropriate Hamilton-Jacobi-Isaacs (HJI) partial differential equation (PDE) [3], [4], [5], [6]. In particular, one can define the game through a value function representing the time-to-capture, with the evader attempting to maximize this function and the pursuers attempting to minimize this function. Under certain technical conditions, the value of the game can be characterized as the solution to an HJI equation, which can be in turn used to synthesize optimal controls for the pursuers minimizing time-to-capture. Solutions to HJI equations are typically found either using the method of characteristics [3], [5], in which optimal trajectories are found by integrating backward from a known terminal condition, or via numerical approximation of the value function on a grid of the continuous state space [6], [7], [8], [9].

The practical usage of the differential game approach, however, faces several computational challenges. While the characteristic solutions are useful in understanding optimal solutions qualitatively, they require backward integration from terminal configurations, which can make it difficult to generate strategies when only the initial configurations of the agents are known. On the other hand, HJI computation on grids suffers from the curse of dimensionality: computing solutions to HJI equations is computationally infeasible for scenarios with large number of agents, as the grid required for approximating the value function grows exponentially in the dimensions of the joint configuration space.

For certain games and game configurations, it is possible to construct strategies for the agents geometrically. For example, pure-distance pursuit, in which a pursuer minimizes the instantaneous distance to the evader, has been shown to be the optimal pursuit strategy for certain zero-sum differential games in open environments [3], as well as a choice of strategy which guarantees capture in simply-connected regions [10]. In other cases, strategies based upon a geometric argument have also been found for coordinating groups of pursuers in open, unbounded spaces [11]. These methods are computationally efficient in generating control strategies, but are often limited

to relatively simple game environments with no obstacles or inhomogeneous speed constraints, such as that occasioned by varying terrain.

When the computation required to find optimal or guaranteed strategies is intractable due to either problem complexity or the need for real-time solutions, a form of model-predictive control (MPC) is often employed. In an MPC formulation, an optimization problem is solved for the control actions of one side (either the pursuers or the evaders), while using a model to predict opponent actions. This solution is implemented for a short time period, and the optimization is then resolved using new agent states. Feedback via this receding horizon implementation is used to correct for errors in the prediction model. This strategy has been used for a number of games, for example in complex pursuit-evasion games such as air combat, in which the roles of the agents may change over time [12], [13]. In general, MPC approaches work best when the predictive model used is a good approximation of the actual strategy of the opponent. When this is true, control inputs can be quickly and efficiently generated for the controlled agents using standard optimization tools. However, proofs of optimality and guarantees of capture for the resulting solutions are often difficult to provide.

In addition to games in continuous time and continuous spaces, research in discrete, turn-based games played on graphs have shown that three pursuers are sufficient and sometimes necessary to capture any evader in a planar graph [14], [15]. These results for discrete games have led to strategies for a class of continuous games known as visibility-based pursuit-evasion [16], [17], [18]. The graph-based analysis has also recently inspired results for continuous games showing that three pursuers are also sufficient and sometimes necessary to capture an evader with equal speed in arbitrary bounded, planar, continuous domains with holes [19], [20], [21]. Similar to the graph and visibility pursuit strategies, they operate on the principle of successively reducing the game domain into a single simply-connected region by having pursuers block the evader from portions of the game space. However, to the best of our knowledge, currently implementable versions of these strategies do not yet exist. For all these methods, the game domain reduction requires searching over a large set of discrete actions, limiting the size of problems that can be practically addressed.

The body of work described in [19], [20], [21] is especially relevant to the multi-agent pursuit-evasion problems discussed in this paper, and is exciting as it provides formal proofs for the existence of pursuit strategies that guarantee capture, with provable bounds on the time-to-capture. On the other hand, the practical computation of such ideal strategies is still a subject of ongoing investigations. Our approach can be viewed as complementary to these theoretical contributions, in that we derive practically computable solution strategies for the particular problem scenario of convex environments and equal speeds. The insights obtained from this scenario may provide the foundation for the construction and implementation of more general strategies.

### III. THE COOPERATIVE PURSUIT PROBLEM

Consider a multi-agent pursuit-evasion game involving  $N$  pursuers and a single evader, taking place in an open, simply connected region  $\Omega$  in  $\mathbb{R}^2$ . Let  $x_e \in \mathbb{R}^2$  be the position of the evader and  $x_p^i \in \mathbb{R}^2$  be the position of pursuer  $i$ . The equations of motion are

$$\begin{aligned} \dot{x}_e &= d, \quad x_e(0) = x_e^0, \\ \dot{x}_p^i &= u_i, \quad x_p^i(0) = x_p^{i,0}, \quad i = 1, \dots, N, \end{aligned} \quad (1)$$

where  $d$  and  $u_i$  are the velocity control inputs of the evader and pursuers, respectively, and  $x_e^0, x_p^{i,0} \in \Omega$  are the initial evader and pursuer positions. The respective agent inputs are constrained to lie within sets  $U_i \subset \mathbb{R}^2$  for the pursuers and  $D \subset \mathbb{R}^2$  for the evader. In this paper,  $U_i$  and  $D$  are assumed to be the following:

$$D = \{d \mid \|d\| \leq v_{e,max}\}, \quad U_i = \{u_i \mid \|u_i\| \leq v_{i,max}\}, \quad (2)$$

where  $v_{e,max}$  and  $v_{i,max}$  are the maximum speeds of the evader and the pursuers, respectively, and  $\|\cdot\|$  denotes the Euclidean norm in  $\mathbb{R}^2$ . The motions of the evader and pursuers, as described by equation (1), are also constrained to lie within the region  $\Omega$ , with

$$x_e(t), \quad x_p^i(t) \in \Omega, \quad \forall t \geq 0. \quad (3)$$

Any velocity input  $d(t)$  or  $u_i(t)$  which satisfies the constraints (2) and (3) is called an admissible input for the evader or pursuer  $i$ , respectively.

The goal of the pursuers is to capture the evader by having at least one of the pursuers bring the evader within a distance  $r_c > 0$  of the pursuer. Let  $C(t)$  be the set of all positions in which the evader is captured at time  $t$ , that is

$$C(t) = \{y \mid \exists i, \|y - x_p^i(t)\| \leq r_c\}$$

The capture condition for the pursuers is then given by

$$x_e(t) \in C(t). \quad (4)$$

To achieve this capture condition, each pursuer selects control inputs using a pursuit strategy  $\mu_i(x_e, x_p^1, \dots, x_p^N)$ , based upon observations of the evader and pursuer positions at each time instant, resulting in the closed-loop system dynamics:

$$\begin{aligned} \dot{x}_e &= d, \quad x_e(0) = x_e^0, \\ \dot{x}_p^i &= \mu_i(x_e, x_p^1, \dots, x_p^N), \quad x_p^i(0) = x_p^{i,0}, \quad i = 1, \dots, N \end{aligned} \quad (5)$$

The evader may use some strategy  $\gamma(x_e, x_p^1, \dots, x_p^N)$  to avoid the pursuers. Any strategy  $\mu_i$  which satisfies the constraints (2) and (3) is called an admissible pursuit strategy for pursuer  $i$ , and similarly for  $\gamma$ . The sets of admissible strategies for the pursuers and the evader are denoted by  $\mathbb{U}$  and  $\mathbb{D}$ , respectively.

A precise statement of the problem for the multi-agent pursuit-evasion game can now be given as the following: for any initial configuration  $x_e^0, x_p^{i,0} \in \Omega$  satisfying  $x_e^0 \notin C(0)$ , find an admissible choice of pursuit strategy  $\mu_i$  for each pursuer  $i$  such that, regardless of any admissible choice of evader input  $d$ , the capture condition (4) is satisfied for some time  $t < \infty$ .

### IV. VORONOI BASED PURSUIT IN CONVEX DOMAIN

We begin our discussion of a pursuit strategy by focusing on an important class of the cooperative pursuit problem with a convex game domain and equal maximum speeds for all the players. Focusing on this special case enables us to develop a scalable and constructive solution to the cooperative pursuit problem, while also providing key insights into the solution of the general cooperative pursuit problem. In the rest of this section, we will first derive the proposed cooperative pursuit strategy and then show that it is guaranteed to capture the evader in finite time under all possible evader strategies. By appropriate re-scaling of problem parameters, our discussion assumes, without loss of generality, that the maximum movement speeds are  $v_{e,max} = v_{i,max} = 1$ ,  $i = 1, \dots, N$ .

#### A. Voronoi-Based Cooperative Pursuit

The pursuit strategy we propose is based on the Voronoi partition of  $\Omega$  generated by the locations of the players. Roughly speaking, the strategy is designed so as to decrease the area of the evader's Voronoi cell over time. Intuitively, as this area decreases towards zero, the capture condition will be satisfied. In this subsection, we will describe the pursuit strategy and some mathematical properties necessary for the proof of finite time capture.

Let  $\mathcal{V}(D) = \{\mathcal{S}_e, V_1, \dots, V_N\}$  be the Voronoi partition of  $D$  generated by the points  $\{x_e, x_p^1, \dots, x_p^N\}$ :

$$\begin{aligned} \mathcal{S}_e &= \{y \in D \mid \|y - x_e\| \leq \|y - x_p^i\|, \forall i \leq N\}, \\ V_i &= \{y \in D \mid \|y - x_p^i\| \leq \min\{\|y - x_e\|, \|y - x_p^j\|\}, \forall j \neq i\}, \quad i \leq N. \end{aligned}$$

Let  $\mathcal{N}_e$  be the set of pursuer indices that are Voronoi neighbors of the evader, with cardinality  $n_e$ . The edge shared by  $\mathcal{S}_e$  and  $V_i$ ,  $i \in \mathcal{N}_e$  is called the *line of control* for pursuer  $i$  and is denoted by  $B_i$ , where  $L_i$  is the length of  $B_i$  (see Figure 1). We denote by  $A$  the area of the Voronoi cell  $\mathcal{S}_e$  containing the evader. This area can be calculated as

$$A(x_e, x_p^1, \dots, x_p^N) = \sum_{j=1}^{n_e} (x_j^e y_{j+1}^e - x_{j+1}^e y_j^e), \quad (6)$$

where  $\{(x_j^e, y_j^e)\}_{j \leq n_e}$  is the set of vertices of  $\mathcal{S}_e$  and  $n_e + 1$  wraps around to the first vertex. It can be easily verified that the area  $A$  depends only on the locations of the neighboring pursuers and that this dependence is smooth whenever the pursuer locations are in  $D$ . The time derivative of  $A$  is given by

$$\frac{dA}{dt} = \frac{\partial A}{\partial x_e} \dot{x}_e + \sum_{i=1}^N \frac{\partial A}{\partial x_p^i} \dot{x}_p^i. \quad (7)$$

Now consider a cooperative pursuit strategy that jointly minimizes  $\frac{dA}{dt}$ . According to (7), this joint objective can be decoupled into the individual objectives of minimizing  $\frac{\partial A}{\partial x_p^i} \dot{x}_p^i$  for each pursuer  $i$ . Since  $A$  depends only on the Voronoi neighbors of the evader, we have  $\frac{\partial A}{\partial x_p^i} = 0$  for all  $i \notin \mathcal{N}_e$ . Thus, for any pursuer  $i$  which is not a Voronoi neighbor of the evader, we simply set  $\mathbf{u}_i = \frac{x_e - x_p^i}{\|x_e - x_p^i\|}$ . On the other hand,

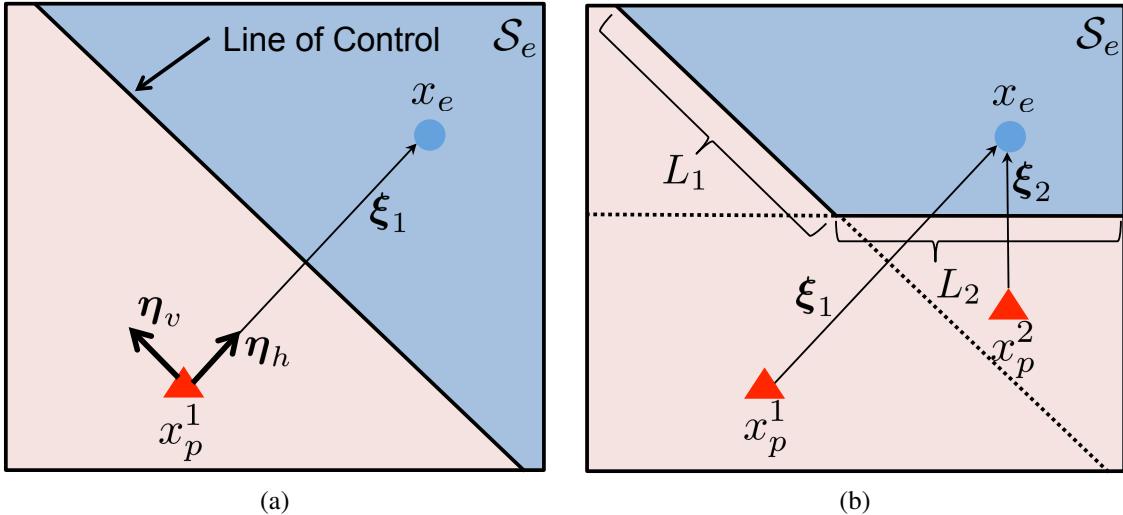


Fig. 1. Illustrations showing the evader's Voronoi cell  $\mathcal{V}_e$  in a convex domain with equal agent speeds (a) for a single pursuer and evader and (b) with an additional pursuer.

for each pursuer  $i \in \mathcal{N}_e$ , the choice of pursuit strategy which minimizes (7) is given by:

$$\mathbf{u}_i^* = \mu_i^*(x_e, x_p^1, \dots, x_p^N) \triangleq \frac{-\frac{\partial A}{\partial x_p^i}}{\left\| \frac{\partial A}{\partial x_p^i} \right\|},$$

where we make use of the assumption that  $v_{max} = 1$ .

To enable analysis of the proposed pursuit strategy, we now derive an analytical expression for  $\mu_i^*$ ,  $i \in \mathcal{N}_e$  using a particular choice of local coordinate system. First, let  $\xi_i(x_e, x_p) = x_e - x_p^i$  be the displacement vector pointing from the location of pursuer  $i$  towards the location of the evader. When there is no ambiguity, its arguments will be omitted and this vector will be denoted simply by  $\xi_i$ . Denote  $\delta_{min}$  to be the minimum of  $\|\xi_i\|$  as  $i$  ranges from 1 to  $N$ . Since  $\delta_{min}(0) > r_c$  until capture, at which time  $\delta_{min}(T) = r_c$ , we have  $\|\xi_i\| \geq r_c$  for all  $i \leq N$  and  $t \in [0, T]$ . Define  $\eta_h^i = \frac{\xi_i}{\|\xi_i\|}$  and let  $\eta_v^i \in \mathbb{R}^2$  be a unit vector orthogonal to  $\eta_h^i$ , as shown in Figure 1(a). The vectors  $\{\eta_h^i, \eta_v^i\}$  define a local coordinate system that depends on the locations of  $x_e$  and  $x_p^i$ . For any  $y \in \mathbb{R}^2$  and  $(x_e, x_p^1, \dots, x_p^N)$  such that  $y + x_p^i \in \Omega$ , define

$$A_i^+(y)|_{(x_e, x_p^1, \dots, x_p^N)} = A(x_e, x_p^1, \dots, x_p^i + y, \dots, x_p^N).$$

Define  $D_h^i A$  and  $D_v^i A$  as the directional derivatives of  $A$  along  $\eta_h^i$  and  $\eta_v^i$ , then

$$\begin{cases} D_h^i A|_{(x_e, x_p^1, \dots, x_p^N)} = \lim_{\epsilon \rightarrow 0} \frac{A_i^+(\epsilon \cdot \eta_h^i)|_{(x_e, x_p^1, \dots, x_p^N)} - A}{\epsilon} \\ D_v^i A|_{(x_e, x_p^1, \dots, x_p^N)} = \lim_{\epsilon \rightarrow 0} \frac{A_i^+(\epsilon \cdot \eta_v^i)|_{(x_e, x_p^1, \dots, x_p^N)} - A}{\epsilon}, \end{cases} \quad (8)$$

where  $A(x_e, x_p^1, \dots, x_p^N)$  is denoted by  $A$  for brevity. From this expression, the partial derivative of  $A$  with respect to  $x_p^i$  is given by

$$\frac{\partial A}{\partial x_p^i} = D_h^i A \cdot \eta_h^i + D_v^i A \cdot \eta_v^i. \quad (9)$$

The geometry of the local coordinate system is illustrated in Figure 2. Recall that  $L_i$  is the length of the line of control  $B_i$ . We denote by  $l_i$  the length of  $B_i$  opposite to  $\eta_v^i$ .

**Lemma 1.** For any  $i \in \mathcal{N}_e$ , it holds that

$$\begin{aligned} D_h^i A &= -\frac{L_i}{2}, \\ D_v^i A &= \frac{l_i^2 - (L_i - l_i)^2}{2\|\xi_i\|}. \end{aligned}$$

*Proof: Perturbation along  $\eta_h^i$ :* A perturbation  $\epsilon$  in the pursuer's position toward the evader moves the line of control  $\xi_i$  toward the evader, and generates a corresponding change in the area of the evader's Voronoi cell  $\delta A_h^i$ , as shown in Figure 2(a). This change in area is

$$\delta A_h^i = -\frac{L_i \epsilon}{2} + O(\epsilon^2),$$

where the  $O(\epsilon^2)$  term depends on the angle of intersection between  $B_i$  and the boundaries of the Voronoi cell  $\mathcal{S}_e$ . From this expression, the directional derivative of  $A$  along  $\eta_h^i$  can be calculated as

$$D_h^i A = \lim_{\epsilon \rightarrow 0} \frac{\delta A_h^i}{\epsilon} = -\frac{L_i}{2}.$$

**Perturbation along  $\eta_v^i$ :** There are two different scenarios for perturbation along  $\eta_v^i$ , corresponding to the two pursuer configurations shown in Figure 1(b). In one case, as for  $x_p^2$  in Figure 1(b),  $\xi_i$  intersects  $B_i$ . A perturbation of  $\epsilon$ , as shown in Figure 2(b), will cause the evader's Voronoi cell to shrink above the new intersection by  $\delta A_{v,1}^i$  and grow below it by  $\delta A_{v,2}^i$ . Let  $\delta A_v^i = \delta A_{v,2}^i - \delta A_{v,1}^i$ , with

$$\begin{aligned} \delta A_{v,1}^i &= \frac{1}{2}((L_i - l_i) - \frac{\epsilon}{2})^2 \frac{\epsilon}{\|\xi_i\|} + O(\epsilon^2), \\ \delta A_{v,2}^i &= \frac{1}{2}(l_i + \frac{\epsilon}{2})^2 \frac{\epsilon}{\|\xi_i\|} + O(\epsilon^2), \end{aligned}$$

where the terms  $O(\epsilon^2)$  again depend on the angle of intersection between  $B_i$  and the boundaries of the Voronoi cell  $\mathcal{S}_e$ .

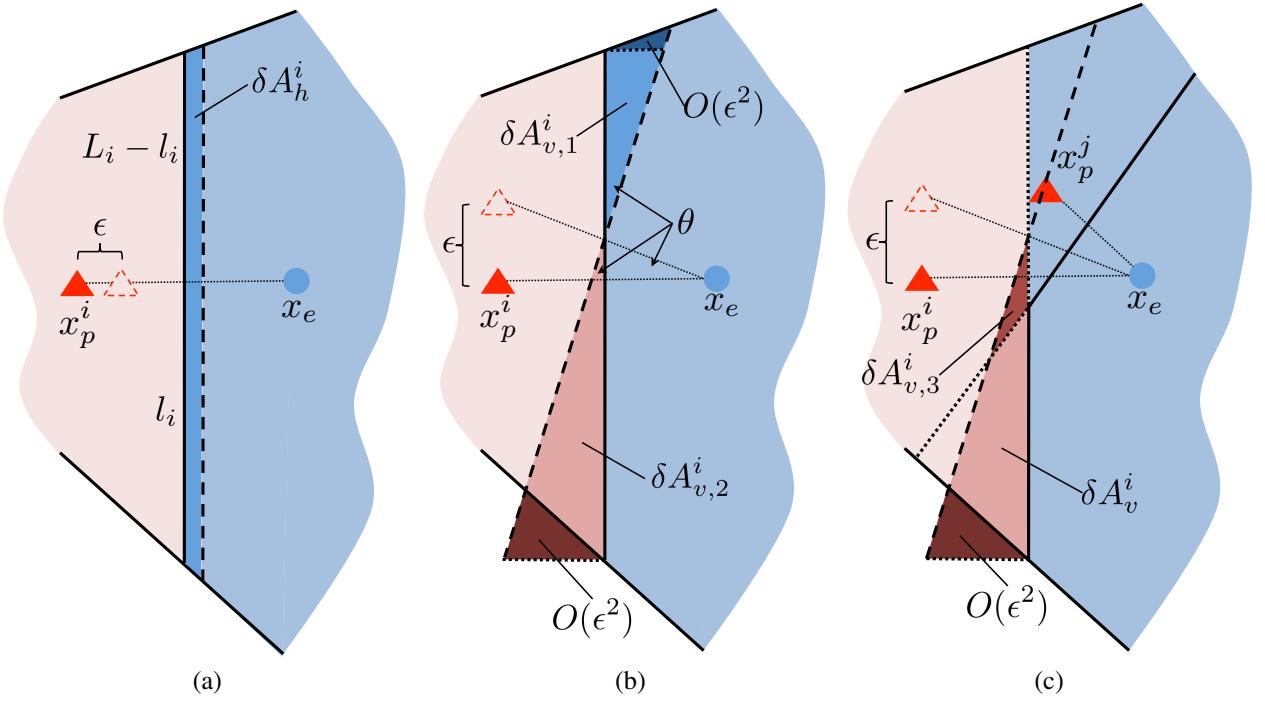


Fig. 2. Variational change in area of the evader's safe-reachable set with respect to (a) a perturbation toward the evader, (b) perturbation parallel to the line of control, and (c) when another pursuer is present and  $\xi_i$  no longer intersects  $B_i$ , as in Figure 1(b).

Thus, the resulting changes in area will be

$$\begin{aligned}\delta A_{v,1}^i &= \frac{(L_i - l_i)^2 \epsilon}{2\|\xi_i\|} + O(\epsilon^2) \\ \delta A_{v,2}^i &= \frac{l_i^2 \epsilon}{2\|\xi_i\|} + O(\epsilon^2)\end{aligned}$$

which implies

$$D_v^i A = \lim_{\epsilon \rightarrow 0} \frac{\delta A_v^i}{\epsilon} = \frac{l_i^2 - (L_i - l_i)^2}{2\|\xi_i\|}.$$

The second case is that of  $x_p^1$  in Figure 1(b), where  $\xi_i$  no longer intersects  $B_i$  due to the presence of other pursuers. As shown in Figure 2(c), the change in area is

$$\delta A_v^i = \delta A_{v,2}^i - \delta A_{v,3}^i,$$

where  $\delta A_{v,2}^i$  is calculated as before and

$$\delta A_{v,3}^i = \frac{1}{2}(l_i - L_i + \frac{\epsilon}{2})^2 \frac{\epsilon}{\|\xi_i\|} + O(\epsilon^2).$$

Note that here  $l_i \geq L_i$ . Letting  $\epsilon \rightarrow 0$  then again it is true that

$$D_v^i A = \lim_{\epsilon \rightarrow 0} \frac{\delta A_v^i}{\epsilon} = \frac{l_i^2 - (l_i - L_i)^2}{2\|\xi_i\|}.$$

■

With the above lemma, the proposed strategy  $\mu_i^*$  can be rewritten in the local coordinate system as

$$\mu_i^* = - \left( \frac{\alpha_h^i}{\sqrt{|\alpha_h^i|^2 + |\alpha_v^i|^2}} \cdot \eta_h^i + \frac{\alpha_v^i}{\sqrt{|\alpha_h^i|^2 + |\alpha_v^i|^2}} \eta_v^i \right), \quad (10)$$

where  $\alpha_h^i$  and  $\alpha_v^i$  are given by

$$\alpha_h^i = -\frac{L_i}{2}, \quad \alpha_v^i = \frac{l_i^2 - (L_i - l_i)^2}{2\|\xi_i\|}.$$

**Lemma 2.** It can be shown that under this choice of pursuit strategy,  $u_i$  always points toward the interior of  $\Omega$ , thus satisfying the constraint from equation (3).

The proof is straightforward but requires some amount of algebra and is thus omitted. Using the above results, the pursuit strategy can be shown to take the following simple form.

**Theorem 3.** Under the proposed pursuit strategy, pursuer  $i$  should always head for the midpoint of the line of control  $B_i$ .

*Proof:* From Lemma 1, the control input is the vector  $(D_h^i A, D_v^i A)$  in the local coordinate system defined by  $\{\eta_h^i, \eta_v^i\}$ . Let  $\alpha$  be the angle between the velocity input and the local horizontal axis defined by  $\eta_h^i$ . It is true that

$$\tan(\alpha) = \frac{D_v^i A}{D_h^i A} = \frac{\frac{l_i^2 - (L_i - l_i)^2}{2\|\xi_i\|}}{-\frac{L_i}{2}} = -\frac{2l_i - L_i}{\|\xi_i\|}; \quad (11)$$

Let  $\beta$  be the angle between the local horizontal axis and the vector from pursuer  $i$  to the midpoint of  $B_i$ . Then

$$\tan(\beta) = \frac{l_i - (L_i/2)}{\|\xi_i\|/2} = -\frac{2l_i - L_i}{\|\xi_i\|}; \quad (12)$$

Therefore  $\alpha = \beta$ , thus establishing the theorem. ■

### B. Proof of guaranteed capture

In this section, we will show that the pursuit strategy outlined above for the convex, equal speed game is guaranteed to capture the evader in finite time, regardless of any admissible

evader input  $d$ . It can be seen that if this holds for the case of a single pursuer ( $N = 1$ ), then the conclusion also extends to the case of multiple pursuers ( $N > 1$ ). Indeed, for the case of  $N > 1$ , one can choose any pursuer  $i$  which is a Voronoi neighbor of the evader and use the arguments for the case of  $N = 1$  to show that the capture condition will be satisfied. This section presents the proof for a single pursuer. Correspondingly, the notation from above will carry through without the indices  $i$ .

First, observe that as  $A$  approaches zero, the evader's Voronoi cell approaches either a line or a point. Either of the two cases clearly implies  $\delta(t) = \|x_e(t) - x_p(t)\| \rightarrow 0$ . The strategy here is then to show that, under the proposed pursuit strategy  $\mu^*$  and any admissible evader control input  $d$ , either the area  $A$  or the distance  $\delta$  is guaranteed to decrease until the capture condition is met.

By way of preliminaries, recall that Lemma 1 implies

$$\frac{\partial A}{\partial x_p} = \alpha_h \boldsymbol{\eta}_h + \alpha_v \boldsymbol{\eta}_v.$$

It can be also verified in a similar manner as the proof of Lemma 1 that the partial derivative  $\frac{\partial A}{\partial x_e}$  in the local coordinate system is given by

$$\frac{\partial A}{\partial x_e} = \alpha_h \boldsymbol{\eta}_h - \alpha_v \boldsymbol{\eta}_v. \quad (13)$$

Also recall that the variable  $L$  in the statement of Lemma 1 depends on the spatial locations of the evader and the pursuer, as well as the geometry of the region  $\Omega$ . For this proof, we make the following definitions of parameters  $l_{\min}$  and  $l_{\max}$ , which depend solely on the geometry of  $\Omega$ :

$$\begin{cases} l_{\min} = \inf_{x_e \in \Omega, x_p \in \Omega} L(x_e, x_p) \\ l_{\max} = \sup_{x_e \in \Omega, x_p \in \Omega} \|x_e - x_p\|. \end{cases} \quad (14)$$

Since  $\Omega$  is bounded and the game ends upon capture, it is true that  $l_{\min} \geq r_c$ ,  $l_{\max} < \infty$ , and  $L \leq l_{\max}$ .

The following result shows that the area  $A$  is always non-increasing under the pursuit strategy  $\mu^*$  for a single pursuer.

**Lemma 4.** *Under the proposed pursuit strategy  $\mu^*(x_e, x_p)$ , the area  $A$  satisfies  $\frac{dA}{dt} \leq 0$  for any admissible evader control input. Furthermore,  $\frac{dA}{dt} = 0$  if and only if the evader follows the following strategy:*

$$\gamma^*(x_e, x_p) = \frac{\alpha_h \boldsymbol{\eta}_h - \alpha_v \boldsymbol{\eta}_v}{\sqrt{\alpha_h^2 + \alpha_v^2}}. \quad (15)$$

*Proof:* For an arbitrary  $d$  with  $\|d\| \leq 1$ :

$$\begin{aligned} \frac{dA}{dt} &= \frac{\partial A}{\partial p} \mu^*(x_e, x_p) + \frac{\partial A}{\partial e} d \\ &= -\sqrt{\alpha_h^2 + \alpha_v^2} + (\alpha_h \boldsymbol{\eta}_h - \alpha_v \boldsymbol{\eta}_v)^T d \leq 0, \end{aligned}$$

where equality holds if and only if  $d(t) = \gamma^*(x_e(t), x_p(t))$ . ■

To prove that the capture condition is achieved in finite time, it is necessary to show that the distance between the pursuer and the evader is strictly decreasing whenever the area  $A$  is constant. For this purpose, define

$$z(x_e, x_p) = \|\xi(x_e, x_p)\|^2 = (x_e - x_p)^T (x_e - x_p).$$

Clearly, the variable  $z$  is the squared Euclidean distance between the evader and pursuer. From the preceding discussions, the range of  $z$  lies in  $[r_c^2, l_{\max}^2]$ . The following result shows that  $\dot{z} < 0$  whenever  $\dot{A} = 0$ .

**Lemma 5.** *If  $\dot{A} = 0$ , then under the pursuit strategy  $\mu^*$ , the following holds:*

$$\frac{dz}{dt} = -\frac{4z}{\sqrt{z + (l_{\max} - l_{\min})^2}} \leq \frac{-4r_c^2}{\sqrt{r_c^2 + l_{\max}^2}}.$$

*Proof:* By Lemma 4,  $\dot{A}(t) = 0$  if and only if  $d(t) = \gamma^*(x_e(t), x_p(t))$ . Thus, if the pursuer input is selected according to the strategy  $\mu^*$ , then whenever  $\dot{A} = 0$ , then

$$\begin{aligned} \dot{z} &= 2(x_e - x_p)^T (x_e - x_p) \\ &= 4\xi^T \left( \frac{\alpha_h}{\sqrt{\alpha_h^2 + \alpha_v^2}} \boldsymbol{\eta}_h \right) \\ &= \frac{-2L\|\xi\|}{\sqrt{\frac{L^2}{4} + \frac{(L^2 - (L-l)^2)^2}{4\|\xi\|^2}}} \\ &= -\frac{4z}{\sqrt{z + (2l - L)^2}} \\ &\leq \frac{-4r_c^2}{\sqrt{r_c^2 + l_{\max}^2}}, \end{aligned}$$

where the second equality follows from the fact that  $\xi^T \boldsymbol{\eta}_v = 0$ , and the last inequality follows from the monotonicity of the function  $\frac{4z}{\sqrt{z + (2l - L)^2}}$  for  $z \geq 0$ . ■

By this result,  $z$  is strictly decreasing whenever the area  $A$  remains constant. However, there remains the possibility that  $z$  is increasing on time intervals where  $A$  is strictly decreasing. The question then becomes whether there exists an evader control that can keep  $z$  inside  $[r_c^2, l_{\max}^2]$  while preventing  $A$  from reaching 0. The following result proves that this is not the case, by exploiting certain properties of the proposed pursuit strategy.

**Lemma 6.** *Under the pursuit strategy  $\mu^*$ , if  $\dot{A} \geq -\beta$  for some positive constant  $\beta > 0$ , then  $\dot{z} \leq -c(\beta)$ , where the bound  $c(\beta)$  is given by*

$$c(\beta) = \frac{\sqrt{2}r_c^2}{l_{\max}} - \frac{4l_{\max}}{l_{\min}}\beta. \quad (16)$$

*Proof:* Under strategy  $\mu^*$ , the following identities hold

$$\begin{cases} \dot{A} = -\sqrt{\alpha_h^2 + \alpha_v^2} + (\alpha_h \boldsymbol{\eta}_h - \alpha_v \boldsymbol{\eta}_v)^T d \\ \dot{z} = 2(x_e - x_p)^T d - \frac{2z}{\sqrt{z + (2l - L)^2}}. \end{cases}$$

Now suppose  $\dot{A} \geq -\beta$ . From the relations  $\boldsymbol{\eta}_h = \frac{x_e - x_p}{\|x_e - x_p\|}$ ,  $\alpha_h = -\frac{L}{2}$ , and  $\alpha_v \boldsymbol{\eta}_v^T d \geq -|\alpha_v|$ , it is true that

$$\begin{aligned} (x_e - x_p)^T d &\leq -\frac{2\|x_e - x_p\|}{L} \left[ \sqrt{\alpha_h^2 + \alpha_v^2} - \beta - |\alpha_v| \right] \\ &\leq \frac{2\|x_e - x_p\|\beta}{L} \leq \frac{2l_{\max}}{l_{\min}}\beta, \end{aligned}$$

which implies that

$$\dot{z} \leq \frac{4l_{\max}}{l_{\min}}\beta - \frac{\sqrt{2}r_c^2}{l_{\max}}.$$

■ Notice that this lemma also implies  $\dot{A} < -\beta$  whenever  $\dot{z} > -c(\beta)$ . Now the previous results in this section will be combined to show that under  $\mu^*$ , the area  $A$  or the distance  $\delta$  decreases until the capture condition is met.

Consider an “energy” function  $E$

$$E = kA + z$$

for a positive constant  $k$  (to be defined subsequently). Clearly,  $E = 0$  if and only if  $A = 0$  and  $z = 0$ , both of which imply that capture occurs. A proof will now be presented that  $E$  will decrease towards zero as  $t$  increases.

**Theorem 7.** *Under the pursuit strategy  $\mu^*$ , if the capture condition has not been achieved before time  $t > 0$ , then for some positive constants  $k, \beta > 0$*

$$\dot{E}(t) \leq E(0) - c(\beta)t$$

where  $E(0)$  is the initial energy and  $c(\beta)$  is defined as in (16).

*Proof:* First, note that the  $\beta$  parameter in Lemma 6 can be chosen such that  $c(\beta) > 0$ . Then, Lemma 5 and 6 together imply that one of the following conditions must be true at any given time:

- 1)  $\dot{A} \geq -\beta$  and  $\dot{z} \leq -c(\beta)$ , or
- 2)  $\dot{z} > c$  and  $\dot{A} < -\beta$ .

Note that the derivative of  $E$  is

$$\dot{E} = k\dot{A} + \dot{z}$$

and  $\dot{A} \leq 0$  for all time. Then, under condition 1,  $\dot{A} \leq 0$  and  $\dot{z} \leq -c(\beta)$ , thus  $\dot{E} \leq -c(\beta)$ . The rate of change of  $z$  is limited by the maximum speed of the two agents and the geometry of the domain. Since  $\dot{z} = 2\delta\dot{\delta}$ ,  $\dot{\delta} \leq l_{max}$ , and  $\dot{\delta} \leq 2$ , then  $\dot{z} \leq 4l_{max}$ . Now, let  $k = \frac{4l_{max} + c(\beta)}{\beta}$ . Under condition 2,  $\dot{A} < -\beta$  and of course  $\dot{z} \leq 4l_{max}$ , thus  $\dot{E} \leq -k\beta + 4l_{max}$ , therefore again  $\dot{E} \leq -c(\beta)$ , guaranteeing that the energy will decrease to 0 in finite time, leading to capture. ■

### C. Discussions

The proposed Voronoi-based pursuit strategy has several distinctive features. First, the strategy adopts a simple analytical expression in terms of the player locations, which enables fast online implementation of the strategy. Next, we have shown that the Voronoi based strategy guarantees capture even with only one pursuer, which distinguishes the proposed strategy from many other heuristic-based pursuit strategies. The strategy also leads to effective cooperation among the pursuers that can dramatically reduce the capture time (see Section VI for numerical illustrations). The reason behind the induced cooperation can be partly seen from the illustration in Fig. 1. Each pursuer affects  $\mathcal{S}_e$  only through its shared boundary with the evader (line of control  $L_1$  and  $L_2$  in the figure). The derived Voronoi based strategy essentially allows each pursuer to push its line of control to the maximal extent along the most effective direction. Finally, the strategy also properly incorporates environment geometries into the control input design for the pursuers. The boundaries of the

game domain provide additional line of controls as if there were more pursuers. Therefore, the pursuit strategy naturally includes constraints created by the environment.

It is worth pointing out that while our analysis does not explicitly account for non-ideal effects such as localization errors and dynamic disturbances, the proposed pursuit strategy provides a degree of robustness to such effects, due to the smooth dependence of the Voronoi area  $A$  on evader and pursuer locations. In particular, for locations where the area function  $A$  is continuously differentiable, small localization errors results in small variations in the pursuer inputs computed using (10). This then translates into small variations in the time derivatives of the area  $A$  and the distance  $z$ , and hence the energy function  $E$ . Thus, if bounds on the localization errors are available, it may still be possible to ensure the asymptotic decrease of the energy function in Theorem 7.

## V. EXTENSIONS TO NONCONVEX DOMAINS WITH UNEQUAL SPEEDS

In this section we extend the area minimization strategy to more realistic pursuit-evasion scenarios with non-convex game domain and unequal maximum speeds. The main challenge lies in the extension of the Voronoi partition concept to handle these realistic conditions. The desired extension requires a definition of Voronoi partitions that is independent of the specific game domain and player dynamics under consideration. Such decompositions in pursuit-evasion games have been considered for non-convex game domains when the pursuers and evader have equal speeds [22] and for convex domains with un-equal speeds [23]. We will examine the combined case of un-equal speeds and non-convex domains. For this purpose, we shall think of the evader’s generalized Voronoi cell as a set of all the points that the evader can reach without being captured by any of the pursuers for all admissible pursuit strategies. We call this set the *Safe Reachable Set* and denote it by  $\mathcal{S}_e$ . The rest of this section will provide a formal definition of the safe reachable set and discuss its numerical computation as well as its applications in cooperative pursuit problems.

### A. Safe-Reachable Set

Given initial conditions  $x_e^0, x_p^{i,0}$ , a point  $y \in \Omega$  is *safe-reachable* if there exists some  $\gamma \in \mathbb{D}$  and  $t \geq 0$  such that  $x_e(t) = y$  and  $x_e(s) \notin C(s)$  for all  $s \in [0, t]$  and all  $\mu_i \in \mathbb{U}$ . The *safe-reachable set*  $\mathcal{S}_e$  of the evader is then defined as

$$\mathcal{S}_e = \{y \in \Omega \mid y \text{ is safe-reachable}\}.$$

Define the minimum time-to-reach function  $\varphi: \Omega \rightarrow \mathbb{R}$  for the evader constrained to  $\mathcal{S}_e$ :

$$\varphi(y) = \min\{t \mid x_e(t) = y, x_e(s) \in \mathcal{S}_e, \forall s \in [0, t]\}. \quad (17)$$

Similarly, a minimum time-to-capture function  $\psi^c: \Omega \rightarrow \mathbb{R}$  for a pursuer in  $\Omega$  is defined as:

$$\psi^c(y) = \min\{t \mid y \in C(t), x_p(s) \in \Omega, \forall s \in [0, t]\}. \quad (18)$$

The minimum time-to-capture function represents, for a point  $y$ , the minimum time required for the pursuer to capture the

evader if the evader were stationary at  $y$ .  $\mathcal{S}_e$  relative to a single pursuer is therefore

$$\mathcal{S}_e = \{y \mid \varphi(y) < \psi^c(y)\}. \quad (19)$$

For multiple pursuers, the evader must reach a point  $y$  before *all* pursuers. Let  $\psi_i^c$  be the minimum time-to-capture function for pursuer  $i$ . The evader must reach the point  $y$  in less time than the *minimum* of all of the time-to-capture functions. Therefore

$$\mathcal{S}_e = \{y \mid \varphi(y) < \psi_i^c(y), \forall i\}.$$

Observe that the definitions of  $\mathcal{S}_e$  and  $\varphi$  are interrelated. Computing  $\mathcal{S}_e$  requires simultaneously computing the set  $\mathcal{S}_e$  and the values of  $\varphi$  within that set. When the pursuer and evader speeds are equal, the safe-reachable set is equivalent to the evader's Voronoi cell in the generalized Voronoi decomposition of the agents [22]. In more complex situations the appropriate time-to-reach values can be computed using modified fast marching methods (FMM) as described in [1]. FMM [24], [25], [26] is a single-pass method used to numerically approximate the *Eikonal equation*, which for the agent dynamics described here can be used to compute the time-to-reach and time-to-capture functions.

It should be noted that  $\mathcal{S}_e$  is defined in the open-loop sense, in that a point in  $\mathcal{S}_e$  can be reached by the evader moving directly toward that point and ignoring the actions of the pursuers. Furthermore, the computational efficiency of the proposed control strategy directly results from the fact that  $\mathcal{S}_e$  is defined in the open-loop sense. On the other hand, the feedback nature of the strategy comes from the receding horizon implementation of the open-loop controls. Namely, the minimization of the open-loop safe reachable set is carried out at each time instant, given the current measurement of system state. The game is still played in the closed-loop sense in that the pursuers react to the movements of the evader (and the evader is presumably reacting to the pursuers).  $\mathcal{S}_e$  is used by the pursuers to guide their pursuit strategy but does not necessarily affect the actions of the evader itself.

### B. Safe-Reachable Set Based Cooperative Pursuit

With the safe reachable set introduced previously, we can now extend the Voronoi-based pursuit strategy to address the general cooperative pursuit problem. We adopt a similar idea that tries to minimize the area of the safe reachable set to progressively limit the evader's future safe locations. With slight abuse of notations, we let  $A$  denote the area of  $\mathcal{S}_e$ . The time derivative of  $A$  remains the same as in (7). A pursuer  $i$  is said to share an *active boundary* with the evader if there is a portion of the boundary of  $\mathcal{S}_e$  where  $\psi_i^c(y) < \psi_j^c(y)$  for all points  $y$  along this portion and all other pursuers  $j$ . In other words, there is a portion of the boundary of  $\mathcal{S}_e$  that is defined purely by the position of pursuer  $i$ . This is analogous to having a shared Voronoi boundary with the evader in a Voronoi decomposition.

Let  $N_e$  be the set of all pursuers that share an active boundary with the evader. The area  $A$  only depends on the pursuers in  $N_e$ , so  $\frac{\partial A}{\partial x_p^i} = 0$  for all  $i \notin N_e$ . Following a

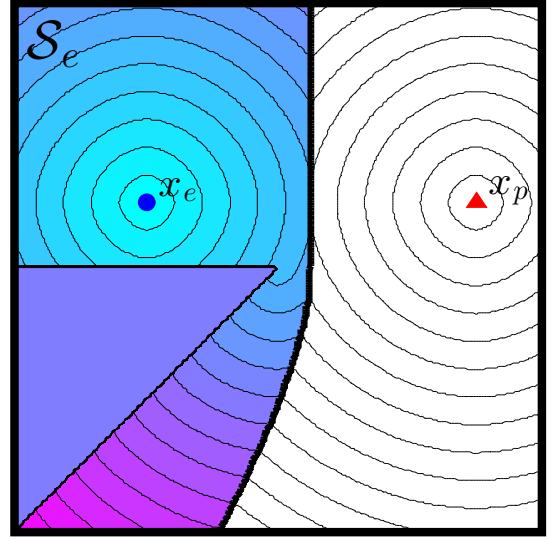


Fig. 3. An example of safe-reachable set computed in a non-convex game domain, with contours plotted for equal time-to-reach values for each agent.

similar idea as in Section IV, any pursuer  $i \notin N_e$  will simply use the pure-distance pursuit strategy. On the other hand, for each pursuer  $i \in N_e$ , the choice of pursuit strategy which minimizes  $\frac{\partial A}{\partial x_p^i}$  is given by:

$$\mathbf{u}_i^* = \mu_i^*(x_e, x_p^1, \dots, x_p^N) \triangleq -v_{i,max} \frac{\frac{\partial A}{\partial x_p^i}}{\left\| \frac{\partial A}{\partial x_p^i} \right\|}.$$

Note that in the case of non-convex domains it may be possible for singular surfaces to arise where multiple inputs result in the same area derivative. In this case the pursuer can simply choose the input which would decrease the distance to the evader the most, as the resulting area change is the same.

While an analytic solution was found for convex domains with equal agents' speeds, it is difficult to construct the safe-reachable set geometrically for the general case. Nevertheless, the modified fast-marching method (FMM) presented in [1] can be used to compute the safe-reachable set on a grid. The area can then be approximated using the grid, allowing numerical computation of the gradient.

The details of the modified FMM algorithm can be found in [1] and will not be repeated here. A sketch of the algorithm is as follows. First, the standard FMM algorithm is used to compute the time-to-capture  $\psi_i^c(y_{j,k})$  for every node  $y_{j,k}$  on the grid. The modified FMM is then used to compute the safe-reachable set  $\mathcal{S}_e$ , beginning with the current position of the evader  $x_e$ , and successively adding points that can be reached in time  $\varphi(y_{j,k})$  with  $\varphi(y_{j,k}) < \psi_i^c(y_{j,k}), \forall i$ . An example of the computation performed for a non-convex region with a triangular obstacle and point capture ( $r_c = 0$ ) is shown in Figure 3, with equal time-to-reach values plotted for the two agents. Note that the boundary of  $\mathcal{S}_e$  is no longer a straight line, since the evader must first move to the corner of the obstacle in order to reach points in the lower portion of the game domain.

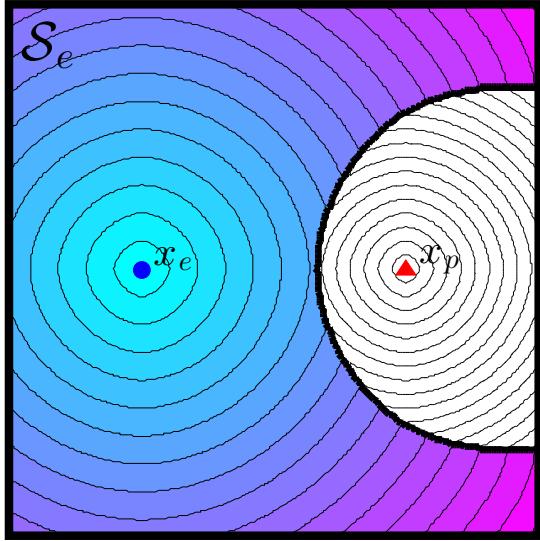


Fig. 4. An illustration of a scenario where the pursuer (red triangle) is slower than the evader (blue circle). Note how  $\mathcal{S}_e$  grows larger as the pursuer gets closer to the evader.

Let  $N_s$  be the number of grid nodes for which  $\varphi(y_{j,k}) < \psi_i^c(y_{j,k}), \forall i$ . The area of  $\mathcal{S}_e$  can be approximated as

$$A \approx N_s h^2$$

where  $h$  is the grid spacing. The gradient with respect to pursuer movements can be numerically approximated by perturbing each pursuer by  $h$  horizontally and vertically on the grid and recomputing  $A$ . The pursuit strategy is identical, with

$$\mathbf{u}_i^* = \mu_i^*(x_e, x_p^1, \dots, x_p^N) \triangleq -v_{i,max} \frac{\frac{\partial A}{\partial x_p^i}}{\left\| \frac{\partial A}{\partial x_p^i} \right\|}.$$

There are several things to note about using safe-reachable area pursuit in non-convex domains with unequal speeds. The first is that the algorithm is only effective in cases where the pursuers are at least as fast as the evader. This is due to the fact that, if the pursuer is slower than the evader, the evader's safe-reachable set may shrink in area as the pursuer goes farther

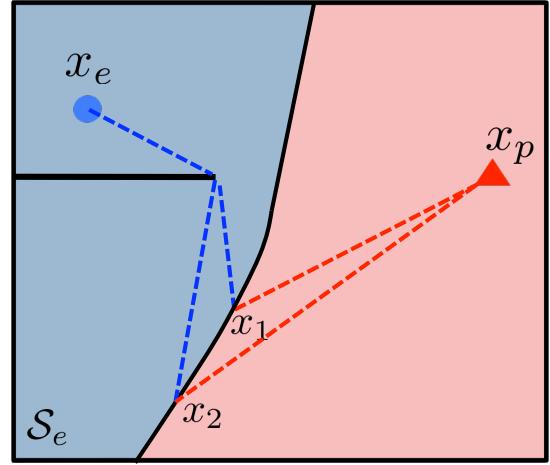


Fig. 5. Illustration of the asymmetry between the evader and the pursuer when the domain is non-convex.

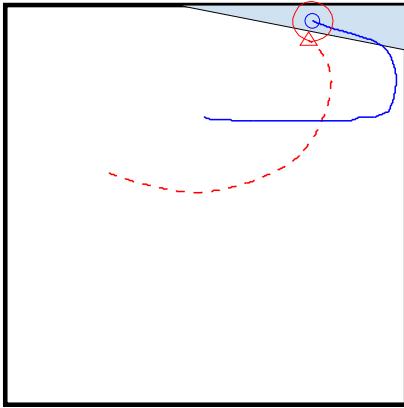
from the evader, as shown in Figure 4. Thus area minimization will likely not lead to capture.

A second point is that currently no proof has been found to guarantee capture in the non-convex case, although the area minimization strategy works well empirically. The major obstacle in this case is that the game is no longer symmetric with respect to the area of  $\mathcal{S}_e$ . This is illustrated in Figure 5, which shows the safe-reachable set for an evader that must turn around a non-convex obstacle. This configuration gives an advantage to the evader, as it moves in the same direction whether it is headed for the point  $x_1$  or  $x_2$ . Thus, if the evader were to move distance  $\epsilon$  toward  $x_1$ , it has also decreased its distance to  $x_2$  by  $\epsilon$ . If the pursuer moves to maintain  $x_1$  on the boundary of the safe-reachable set, then it can move  $\epsilon$  toward  $x_1$ , but it will have moved some distance less than  $\epsilon$  toward  $x_2$ , and  $x_2$  will have become part of the evader's safe-reachable set. In this manner, the evader is able to increase the area of its safe-reachable set regardless of the pursuer input.

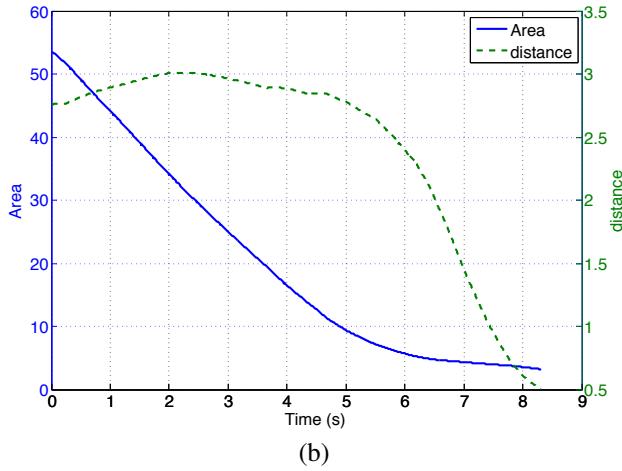
However, an advantage of this approach is that the safe-reachable set is defined without explicit dependence on the dynamics of the agents. For example, although simple kinematic dynamics are considered here, it is possible to use FMM formulations to compute reachable sets for more complex dynamics such as the Dubin's car [27]. The computational approach also lends itself well to cases with spatially-varying velocity profiles, where the maximum speeds of the agents may change depending on position [1]. Thus the pursuit algorithm can be used whenever the safe-reachable set of the evader can be computed in an efficient manner.

## VI. SIMULATION RESULTS

A number of simulations were conducted to evaluate the proposed safe-reachable area minimization pursuit strategy. The performance of the pursuit strategy was compared in a sequence of trials against two other pursuit algorithms: pure-distance pursuit, where each pursuer instantaneously minimizes the distance between itself and the evader, and a numerical approximation of the optimal Hamilton-Jacobi-Isaacs solution on a grid. The pure-pursuit strategy was chosen



(a)



(b)

Fig. 6. Simulation results for a single pursuer (triangle, dashed line) and evader (circle, solid line), showing (a) the trajectories and (b) the area of  $S_e$  and the distance between the agents over time.

for comparison purposes because it is straightforward to implement on arbitrarily defined game domains and is the optimal pursuit strategy for open domains with no boundaries. It is also guaranteed to result in capture for closed, simply connected domains [10]. Several simulation examples will be used to highlight some qualitative properties of the safe-reachable area-minimization pursuit strategy, and then the quantitative results of the numerical trials will be presented.

#### A. Illustrative Examples

A few simulations are presented here to highlight some qualitative aspects of the safe-reachable area-minimization pursuit algorithm. The first set of simulations were conducted in a  $10 \times 10$  square using the convex, analytic pursuit algorithm. The maximum speed was 1 for all agents, with capture radius of 0.5, and time steps of 0.01. In these simulations the trajectory of the evader was controlled by human input, and pursuers that did not have a line of control bordering on the evader's safe-reachable set were commanded to head straight for the evader.

An example trajectory for a game with a single pursuer is shown in Figure 6. The critical trade-off between area and distance is highlighted by this example. Note that initially the pursuer did not move directly toward the evader, and thus the

distance between the agents did not decrease, but the area decreased very quickly. Near the end of the game the area decreased slowly while the distance decreased very quickly.

Figure 7 shows a comparison between the pure-distance pursuit strategy and safe-reachable area-minimization pursuit for a scenario with 3 pursuers and highlights the cooperation in this pursuit strategy. Pure-distance pursuit is shown in Figure 7(a), and safe-reachable area-minimization pursuit is shown in Figure 7(b). The pursuers began closely grouped, and in pure-distance pursuit they acted independently, resulting in a prolonged chase. With the safe-reachable area pursuit strategy, the pursuers gradually separated to surround the evader. The cooperative behavior effectively contained the evader, limiting its movements until capture was achieved.

Pursuit in a non-convex environment is shown in Figure 8, which shows 2 pursuers (red triangles) pursuing a single evader (blue circle) in a simple, non-convex region with a triangular obstacle. The evader's safe-reachable set  $S_e$  is shown at each time, with equal time-to-reach contours plotted within  $S_e$ .

The simulations were conducted in Matlab on a Macbook Pro laptop with a 2.2 Ghz Intel Core i7 processor with 8 GB of RAM, with computation per time-step of less than 1ms to calculate inputs for all the pursuers in the analytic, convex case, and about 100ms for each pursuer using FMM. Note that some small errors are introduced by discretization of the control scheme when distances between the evader and pursuers are comparable to the distance traveled by an agent in a single time step. Reducing the time step alleviates the problem without eliminating it entirely, and increasing the capture radius also reduces the chance of this problem occurring. It is possible that some relationship can be found between step size, velocity, and the capture radius to formally guarantee this in a discrete-time situation.

#### B. Comparison Tests

The results of the comparison tests conducted to evaluate the safe-reachable area-minimization pursuit strategy will now be presented. Two groups of trials were conducted. The first set of trials matched safe-reachable area-minimization pursuit, pure-distance pursuit, and the numerical HJI strategy against each other in tests with one pursuer and one evader. For these tests, a numerical approximation to the HJI equation was computed on a  $40 \times 40$  grid for a simple non-convex region, shown in Figure 9(b), and the pursuer and evader strategies were evaluated by numerical differentiation. The three different pursuit strategies were then evaluated against the approximate optimal evader strategy for 500 initial conditions generated randomly.

The results of this test are displayed in Figure 10. In general, the area-minimization strategy performed slightly worse than the numerical HJI pursuit strategy, and the pure-distance pursuit strategy performed typically the worst among the three. However, it should be noted that the numerical HJI strategy depends on numerical differentiation of the approximated value function on a grid, and numerical errors can lead to sub-optimal pursuer and evader actions. For example, in 29%

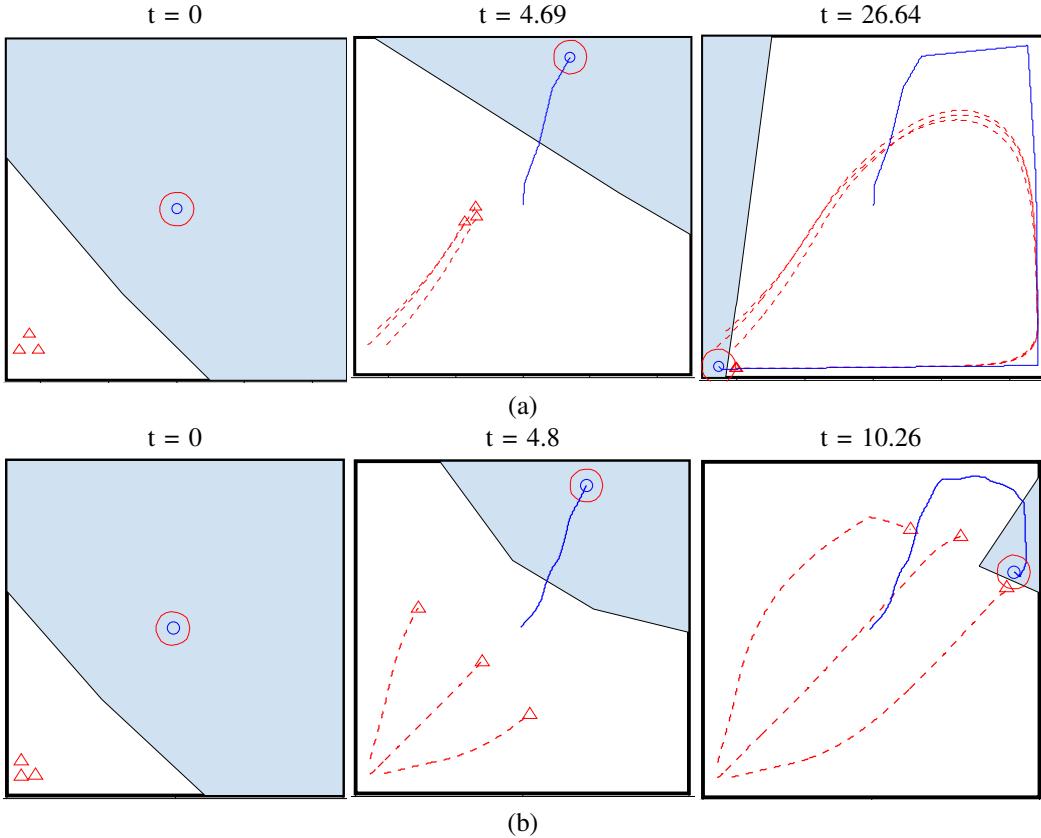


Fig. 7. A scenario with 1 evader (blue circle) and 3 pursuers (triangles, dotted lines) using (a) pure-distance pursuit, and (b) safe-reachable area minimization, highlighting the cooperation enforced by the safe-reachable area minimization pursuit strategy when the pursuers begin tightly spaced.

of trials the area minimization pursuit strategy out-performed the “optimal” HJI pursuer, and similarly in 8.5% of trials the pure-distance pursuit strategy resulted in faster capture-times.

In addition to the numerical issues discussed above, due to computational complexity, the HJI solution can only be found for the case of a single pursuer. To evaluate the performance of the pursuit strategy with multiple pursuers, a further series of tests were conducted comparing safe-reachable area-minimization pursuit with pure-distance pursuit. For these tests, the evader strategy was defined as the following: the evader selects as its target point  $y^*$  the farthest point from itself in  $\mathcal{S}_e$ , that is

$$y^* = \max_{y \in \mathcal{S}_e} \delta_g(y, x_e)$$

where  $\delta_g$  is the geodesic distance between  $y$  and  $x_e$ . Once the evader reaches a certain distance (set here as half of the capture radius) from  $y^*$ , a new target is selected and the evader will proceed toward this target.

Three sets of trials were conducted with 1, 2, and 3 pursuers, with one set in a square, convex region using the analytically derived pursuit strategy and two others in non-convex regions using FMM, shown in Figure 9. For each set, 500 random initial sets of pursuer and evader positions were generated. The results of the tests are summarized in Figure 11, showing histograms of the difference in capture-times between trials, defined as the capture-time required for the safe-reachable area strategy minus the capture-time required for pure-distance pursuit given a set of initial conditions.

Table I shows the fraction of trials in each case where the area pursuit strategy resulted in faster capture times than pure-distance pursuit. Figure 11(a) shows the distribution of results in the convex environment. In this scenario, the safe-reachable area pursuit strategy resulted in clearly superior performance, with the vast majority of trials resulting in faster capture times. The distribution of times seems somewhat bimodal in these trials, with a number of trials where safe-reachable area minimization pursuit and pure-distance pursuit performed similarly, and then a large group where the safe-reachable area minimization pursuit strategy was clearly superior.

The results for the non-convex scenarios are shown in Figure 11(b) for the simple non-convex environment, and in Figure 11(c) for the complex non-convex environment. The simple non-convex case still resulted in a large majority of trials where the safe-reachable area minimization pursuit strategy gave faster capture times, although in a smaller percentage of trials than the convex scenario. The complex non-convex case showed a decline in the performance of the safe-reachable area minimization strategy relative to the pure-distance pursuit strategy. This is due to the fact that the obstacles create areas where the width of the free space is of similar size to the capture radius, thus the pure-distance pursuit can still “trap” the evader, lessening the advantage conferred by the safe-reachable area strategy. In fact, it is to be expected that as the space becomes more and more similar to a single long, narrow corridor, the pure-distance pursuit and area-pursuit strategies should have identical performance. This

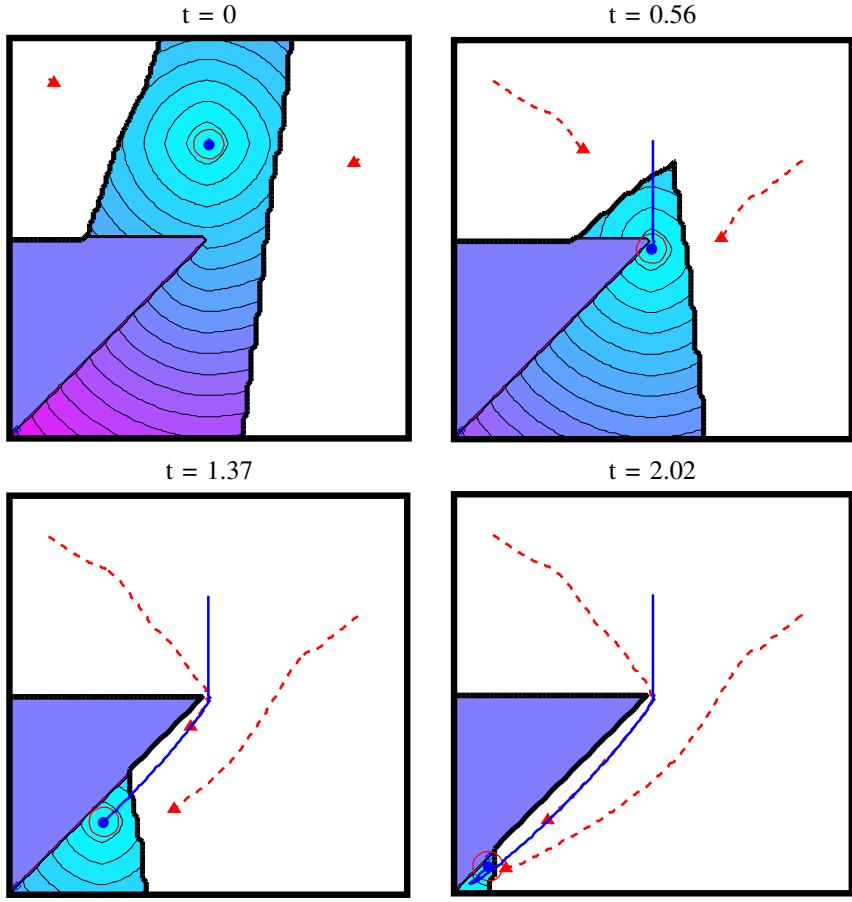


Fig. 8. An example scenario showing 2 pursuers and 1 evader in a non-convex environment, solved using the modified FMM algorithm.

	1 pursuer	2 pursuers	3 pursuers
Convex	91.6%	98.2%	96.6%
Simple Non-convex	67.2%	86.2%	86.4%
Complex Non-convex	34.0%	52.8%	60.4%

TABLE I

PERCENTAGE OF TRIALS FOR WHICH THE SAFE-REACHABLE AREA  
MINIMIZATION PURSUIT STRATEGY OUT-PERFORMED THE  
PURE-DISTANCE PURSUIT STRATEGY.

is especially evident in the trials with only 1 pursuer, where only 34% of trials resulted in faster capture with safe-reachable area-minimization pursuit, with a long tail of trials where the area pursuit performed much worse than pure-distance pursuit. These typically occurred in trials where the evader was able to escape from a confined portion of the game domain against the safe-reachable area minimizing pursuer, in part due to numerical errors in the area differentiation. Nonetheless, the safe-reachable area-minimization strategy showed a clear superiority in the trials with 2 and 3 pursuers, demonstrating the benefit of cooperation.

## VII. EXPERIMENTAL RESULTS

Experiments were conducted using the safe-reachable area minimization pursuit strategy on the BErkeley Autonomy and Robotics in CApture-the-flag Testbed (BEARCAT). BEARCAT is a novel testbed consisting of smartphones

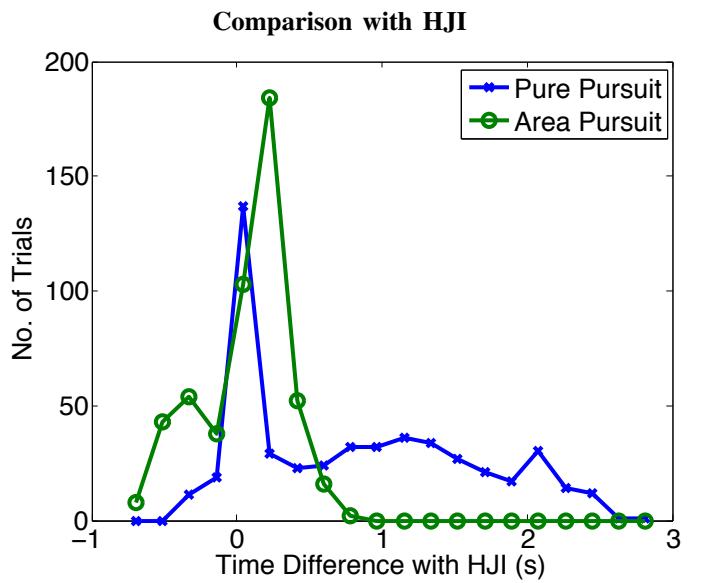


Fig. 10. Histogram showing the difference in capture times in each single pursuer trial between the safe-reachable area minimization strategy, the pure-distance pursuit strategy, and the numerical Hamilton-Jacobi-Isaacs pursuit strategy.

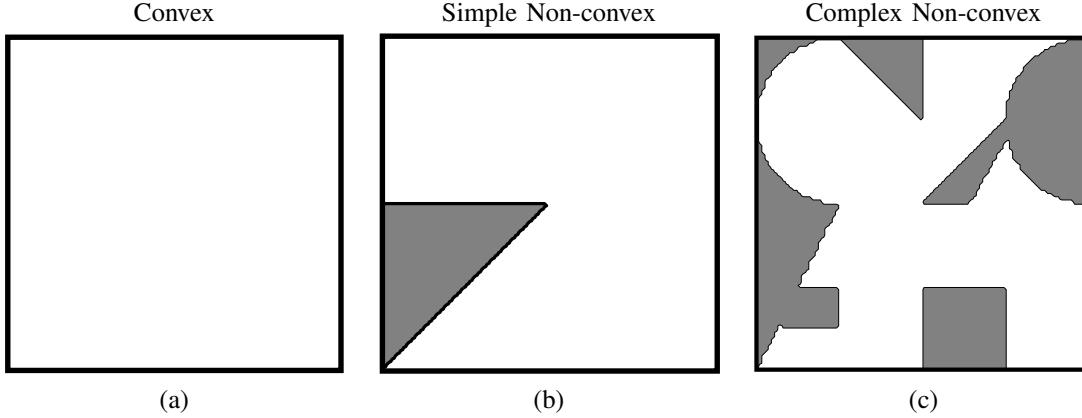


Fig. 9. Domains used for the comparison simulations.

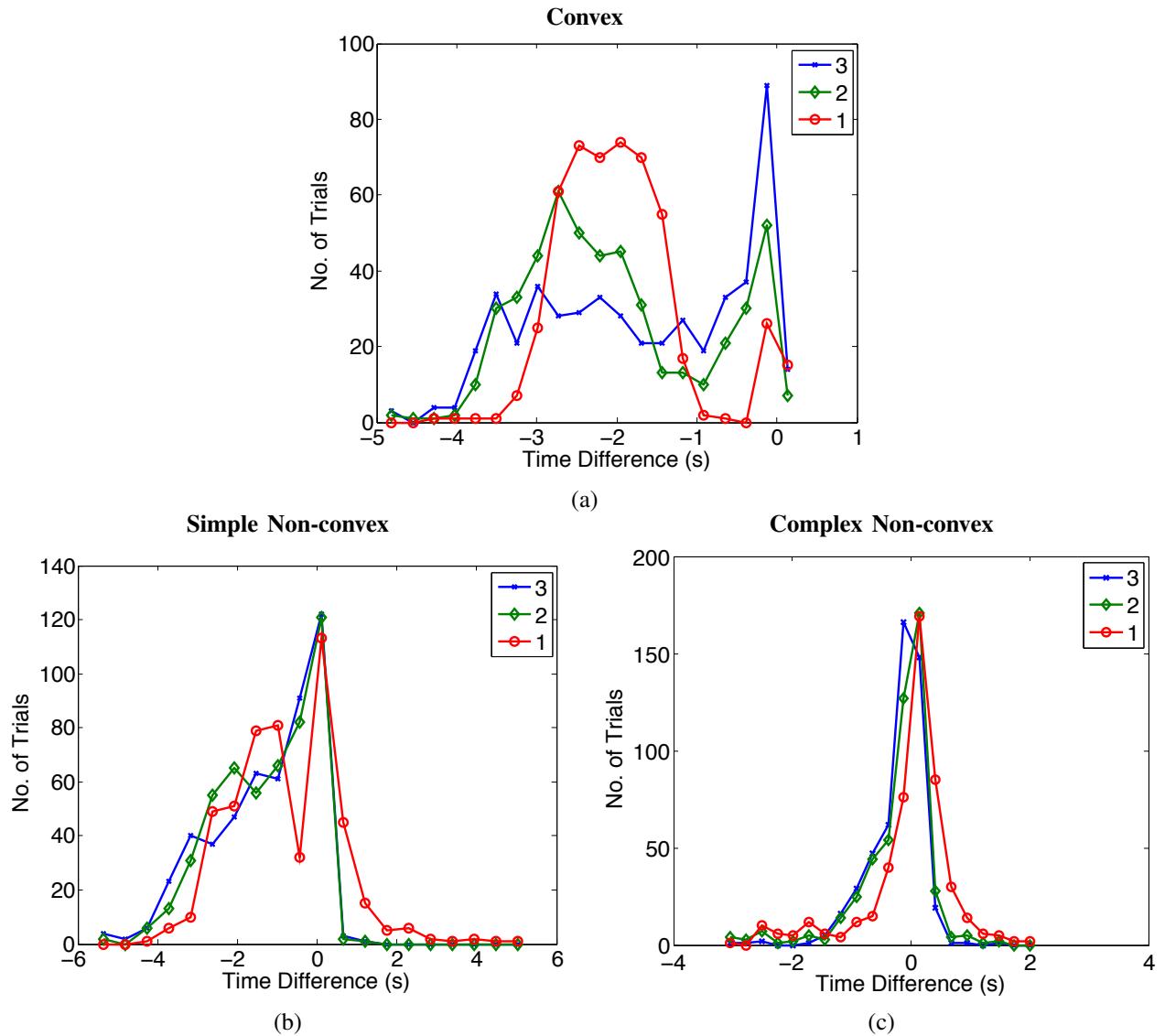


Fig. 11. Histograms showing the number of trials versus difference in capture time between the pure-distance pursuit and safe-reachable area minimization strategies for (a) the convex game domain, (b) the simple non-convex domain, and (c) the complex non-convex domain.

connected to off-board computation and quadrotor UAVs. It is designed for research into automated assistance for human agents in adversarial games. The primary purpose of BEARCAT is to provide a flexible testbed where human agents can participate in an adversarial game such as capture-the-flag, while receiving guidance from computational tools and working with autonomous agents such as UAVs. The testbed allows experiments to be performed by playing reach-avoid and pursuit-evasion games on the Berkeley campus. The major components of BEARCAT are illustrated in Figure 12, consisting of HTC Incredible smartphones [28], laptop computers, and quadrotor UAVs. The different components are networked via wireless communications, providing a complete system for both tracking agents and providing them with access to resources such as computed game solutions and UAV sensing.

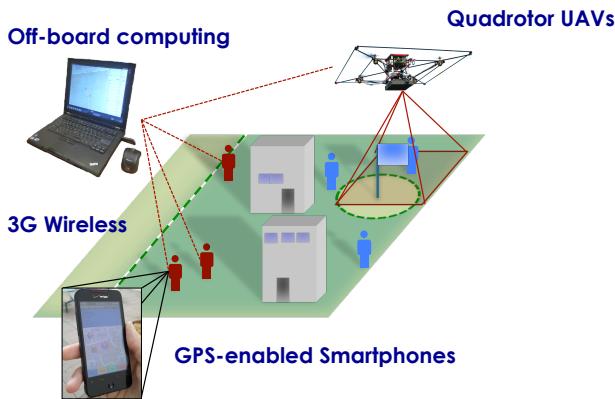


Fig. 12. Components of the BEARCAT experimental platform.

The purpose of the tests was to evaluate the feasibility of the pursuit strategy as a tool for guiding human agents in a cooperative pursuit task. For these experiments, the UAV component of BEARCAT was not utilized. The game was played by two pursuers and a single evader in a small, convex field measuring about 80m x 40m. Agents were tracked using HTC Incredible smartphones, with the agent positions, game region, and the evader safe-reachable set also displayed on the phones. Pursuer strategies were computed using the analytic formulation directly on the phones, and the optimal heading direction was displayed for each pursuer, along with the pursuer's active boundary of the safe-reachable set.

Results from one of these experiments is shown in Figure 13. In this experiment, the pursuers began the game on the western side of the field, with the evader on the eastern side. The camera was placed to the north of the field, looking south. The pursuers were able to successfully trap and capture the evader.

An important point to note is that during these experiments, the GPS positions of the agents were not always perfectly reliable, and as a result the optimal headings were not always correct. Nonetheless, the pursuers were able to use the boundaries of the evader safe-reachable set to guide their actions. Instead of following the optimal headings exactly, each pursuer's active boundary of  $\mathcal{S}_e$  gave an idea of that pursuer's area of responsibility, such that each pursuer could

still reduce the area of  $\mathcal{S}_e$  by attempting to "push" the boundary toward the evader by moving toward the perceived midpoint of the boundary. In addition, the visual display of the safe-reachable set allowed for implicit cooperation between the human pursuers, without the need for verbal communication. Thus, even though the agents were unable to always directly use the computed optimal headings, the safe-reachable set still enabled the pursuers to efficiently coordinate in capturing the evader.

## VIII. CONCLUSIONS AND FUTURE WORK

The pursuit strategy presented in this work has several important strengths. The construction of the Voronoi partition or safe-reachable set allows a high-dimensional problem to be reduced to lower dimensions, easing the computational burden. Additionally, each pursuer can compute its inputs independently, allowing the strategy to run efficiently in real time. Yet information is shared through the partition itself, enabling cooperation between the pursuers and reducing capture time. The Voronoi partition also encompasses global information about the game domain, giving an advantage over strategies such as pure-distance pursuit that ignores the presence of obstacles and boundaries.

One weakness of the strategy as presented is that it cannot handle more general domains that are not simply connected, where obstacles form holes in the domain. Further work is required to adapt the concepts presented here to more general domains. In addition, the numerical evaluation of the area gradient may also be improved. From the simulations and experiments it was noted that the behavior of the pursuers was typically smoother with the analytic input formula in convex domains than with the FMM computation.

Overall, safe-reachable area minimization seems to hold promise as a cooperative pursuit solution approach. As seen before in other works [29], [9], the visualization of the reachable set is a useful tool for human agents. Although GPS noise and time delay rendered the optimal headings sometimes unreliable in practice, the visualization of the set itself allowed the agents to compensate for these disturbances and head in the correct general direction to capture the evader. This result supports the idea that reachable sets are effective visual tools for assisting human agents.

## ACKNOWLEDGMENTS

The authors would like to thank Ryo Takei for his help in using the modified FMM to perform the safe-reachable set computations presented in this paper. In addition, they would like to thank Scott Hoag, Steven Xu, Ayushi Samaddar, and Catherine Pavlov for their contributions in developing the BEARCAT experimental platform and the subsequent experiments. This work would not have been possible without their assistance and support.

## REFERENCES

- [1] R. Takei, H. Huang, J. Ding, and C. Tomlin, "Time-optimal multi-stage motion planning with guaranteed collision avoidance via an open-loop game formulation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, 2012.

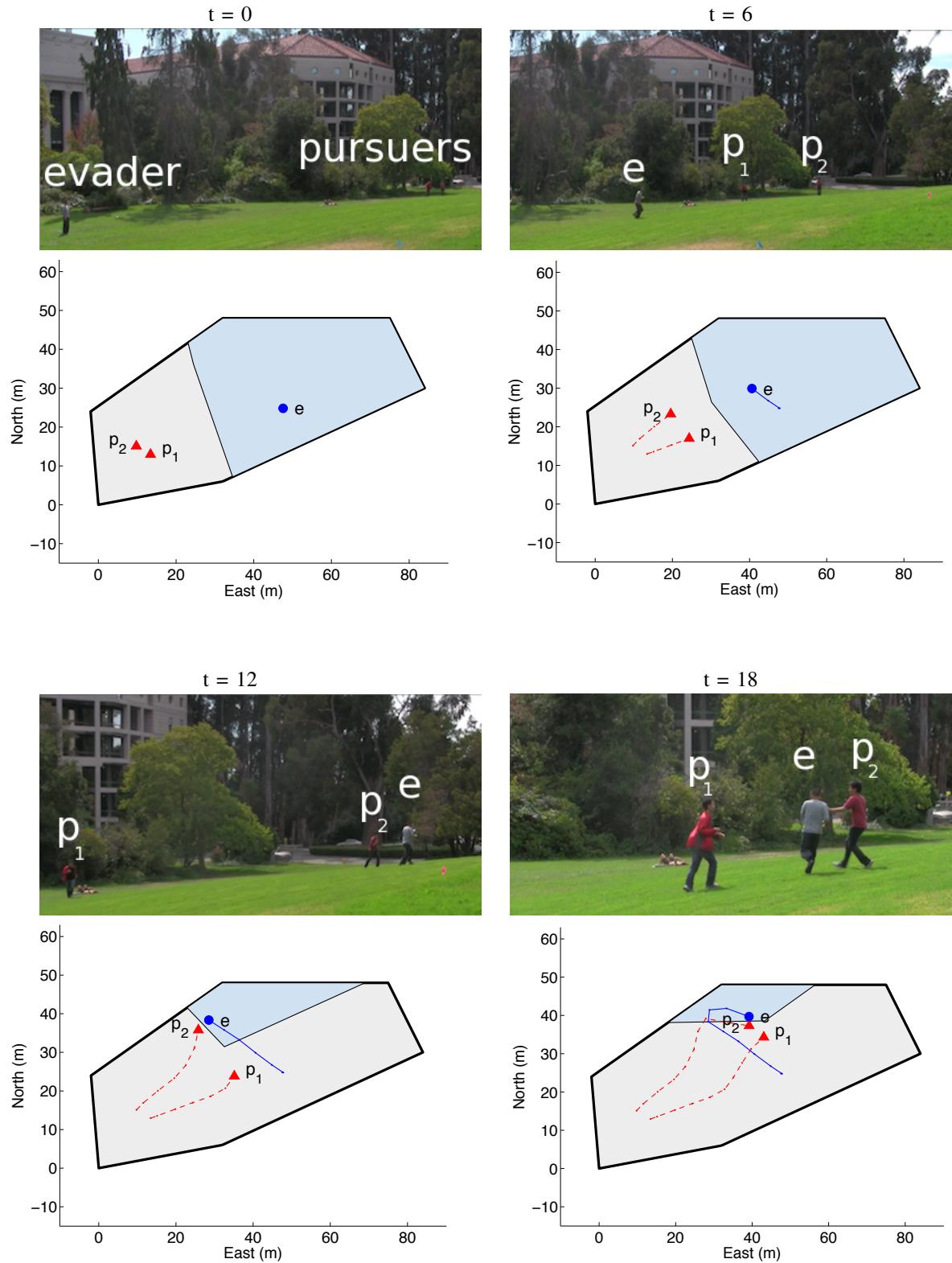


Fig. 13. Video stills and data plotted for an experiment using the safe-reachable area minimization pursuit strategy in BEARCAT. For visual clarity, the evader is labeled **e** and the pursuers are labeled **p<sub>1</sub>** and **p<sub>2</sub>**.

- [2] H. Huang, W. Zhang, J. Ding, D. Stipanović, and C. Tomlin, “Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers,” in *Proceedings of the IEEE International Conference on Decision and Control*, Orlando, Florida, 2011.
- [3] R. Isaacs, *Differential Games*. New York: Wiley, 1967.
- [4] L. C. Evans and P. E. Souganidis, “Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations,” *Indiana University Mathematics Journal*, vol. 33, no. 5, pp. 773–797, 1984.
- [5] T. Başar and G. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA: SIAM, 1999.
- [6] I. Mitchell, A. Bayen, and C. Tomlin, “A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games,” *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.
- [7] M. Falcone and R. Ferretti, “Semi-Lagrangian schemes for Hamilton-Jacobi equations, discrete representation formulae and Godunov methods,” *Journal of Computational Physics*, vol. 175, no. 2, pp. 559–575, 2002.
- [8] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, “Reachability calculations for automated aerial refueling,” in *Proceedings of the IEEE International Conference on Decision and Control*, Cancun, Mexico, 2008, pp. 3706–3712.
- [9] H. Huang, J. Ding, W. Zhang, and C. Tomlin, “A differential game approach to planning in adversarial scenarios: a case study on capture-the-flag,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.
- [10] S. Alexander, R. Bishop, and R. Ghrist, “Pursuit and evasion in non-convex domains of arbitrary dimensions,” in *Proceedings of the Robotics: Science and Systems Conference*, Philadelphia, Pennsylvania, August 2006.
- [11] S. Kopparty and C. Ravishankar, “A framework for pursuit evasion games in  $R^n$ ,” *Information Processing Letters*, vol. 96, no. 3, pp. 114–122, 2005.
- [12] J. Sprinkle, J. Eklund, H. Kim, and S. Sastry, “Encoding aerial pursuit/evasion games with fixed wing aircraft into a nonlinear model predictive tracking controller,” in *Proceedings of the IEEE International Conference on Decision and Control*, Atlantis, Bahamas, December 2004.
- [13] J. McGrew, J. How, L. Bush, B. Williams, and N. Roy, “Air combat strategy using approximate dynamic programming,” *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, Aug 2008.
- [14] T. Parsons, “Pursuit-evasion in a graph,” *Theory and Applications of Graphs*, pp. 426–441, 1978.
- [15] M. Aigner and M. Fromme, “A game of cops and robbers,” *Discrete Applied Mathematics*, vol. 8, no. 1, pp. 1–12, 1984.
- [16] L. Guibas, J.-C. Latombe, S. LaValle, D. Lin, and R. Motwani, “Visibility-based pursuit-evasion in a polygonal environment,” in *Algorithms and Data Structures*, ser. Lecture Notes in Computer Science, F. Dehne, A. Rau-Chaplin, J.-R. Sack, and R. Tamassia, Eds. Springer Berlin / Heidelberg, 1997, vol. 1272, pp. 17–30.
- [17] S. LaValle and J. Hinrichsen, “Visibility-based pursuit-evasion: the case of curved environments,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, pp. 196–202, April 2001.
- [18] B. P. Gerkey, S. Thrun, and G. Gordon, “Visibility-based pursuit-evasion with limited field of view,” *International Journal of Robotics Research*, vol. 25, no. 4, pp. 299–315, 2006.
- [19] D. Bhaduria, K. Klein, V. Isler, and S. Suri, “Capturing an evader in polygonal environments with obstacles: The full visibility case,” *Int. J. Rob. Res.*, vol. 31, no. 10, pp. 1176–1189, Sep. 2012. [Online]. Available: <http://dx.doi.org/10.1177/0278364912452894>
- [20] Z. Zhou, J. Shewchuk, H. Huang, and C. Tomlin, “Smarter lions: Efficient full-knowledge pursuit in general arenas, i: Simply connected domains,” in *IEEE Conference on Decision and Control*, Florence, Italy, 2013.
- [21] ———, “Smarter lions: Efficient full-knowledge pursuit in general arenas, ii: Domains with obstacles,” in *IEEE Conference on Decision and Control*, submitted, Florence, Italy, 2013.
- [22] W. Cheung, “Constrained pursuit-evasion problems in the plane,” Master’s thesis, Faculty of Graduate Studies, Computer Science, University of British Columbia, 2005.
- [23] E. Bakolas and P. Tsotras, “Optimal pursuit of moving targets using dynamic voronoi diagrams,” *Proceedings of the IEEE International Conference on Decision and Control*, 2010.
- [24] J. A. Sethian, “Fast marching methods,” *SIAM Review*, vol. 41, no. 2, pp. 199–235, 1999.
- [25] ———, *Level set methods and fast marching methods*, 2nd ed., ser. Cambridge Monographs on Applied and Computational Mathematics. Cambridge, UK: Cambridge University Press, 1999, vol. 3.
- [26] M. Falcone, “Fast marching methods for front propagation,” November 2007, lecture notes at “Introduction to Numerical Methods for Moving Boundaries”.
- [27] R. Takei and R. Tsai, “Optimal trajectories of curvature constrained motion in the hamilton-jacobi formulation.” *J. Sci. Comput.*, vol. 54, no. 2-3, pp. 622–644, 2013. [Online]. Available: <http://dblp.uni-trier.de/db/journals/jscic/jscic54.html#TakeiT13>
- [28] HTC, “HTC Droid Incredible,” 2012, <http://www.htc.com/us/products/droid-incredible-verizon>.
- [29] M. Oishi, I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, “Invariance-preserving abstractions of hybrid systems: Application to user interface design,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 229–244, 2008.