

Hybrid Control for Visibility-Based Pursuit-Evasion Games

Volkan Isler

GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104, USA
isleri@cis.upenn.edu

Calin Belta

Mechanical Engineering and Mechanics
Drexel University
Philadelphia, PA 19104, USA
Email: calin@drexel.edu

K. Daniilidis and G. J. Pappas

GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104, USA
{kostas, pappasg}@cis.upenn.edu

Abstract—Pursuit-evasion games in complex environments have a rich but disconnected history. Continuous or differential pursuit-evasion games focus on optimal control methods, and rely on very intense computations in order to provide locally optimal controls. Discrete pursuit-evasion games on graphs are algorithmically much more appealing, but completely ignore the physical dynamics of the players, resulting in possibly infeasible motions. In this paper, we present a provable and algorithmically feasible solution for visibility-based pursuit-evasion games in simply-connected environments, for players with dynamic constraints. This is achieved by combining two recent but distant results.

I. INTRODUCTION

In pursuit-evasion games, a pursuer tries to capture an evader who, in turn, actively tries to avoid capture. Designing pursuit strategies is a fundamental challenge in robotics that has many applications. For example, in the well-known homicidal chauffeur game, a driver wants to collide with a pedestrian and the goal is to determine conditions under which he can (not) do so. Among the numerous applications of this game are missile guidance, collision avoidance and air traffic control (cf. [4]).

Early work on pursuit evasion games focused on simple environments [9]. However, many robotics applications (e.g. surveillance) leads to formulations of pursuit-evasion games that take place in complex environments [22].

Historically, there have been two approaches for studying pursuit evasion games. On one hand, there are continuous games that explicitly model the physical motion and constraints of the players [9], [14], [4]. Even though this is a mature area, optimality results are typically local, and searching for optimal control inputs rely on very expensive numerical solutions of Hamilton-Jacobi-Isaacs partial differential equations.

In complicated environments with many players, this approach faces serious scalability challenges.

On the other hand, algorithmic approaches for purely discrete games come equipped with theoretical results which give global guarantees. These games are either played in a purely discrete environment such as a graph [2], [16], [15], [1], [6], [2], [11], or in a continuous environment without any motion constraints [7], [19], [21], [13], [20], [18], [12], [10]. These models of the players abstract away the physical dynamics and constraints of motion. This may result in purely discrete strategies which are dynamically infeasible.

In this paper, we initiate a study that attempts to combine discrete and continuous approaches. We focus on the visibility-based pursuit-evasion game introduced in [21], [7], but for players with physical dynamics. In this game, the goal is to locate an unpredictable and adversarial evader hiding inside a polygonal environment. Recently, it has been shown that [10] there exists a randomized strategy for a single pursuer to locate the evader in any simply-connected environment – even if the evader is arbitrarily faster than the pursuer, knows the position of the pursuer at all times, and actively avoids capture. This randomized strategy is based on triangulations of environment, and hence is compatible with the environment where the game is played. However, the discrete strategy may not be compatible with the dynamic model of the pursuer. This may result in winning (discrete) strategies with infeasible physical (continuous) implementations.

Refining strategies from the discrete to the continuous world has received much attention in the hybrid systems community [3]. In this paper, we utilize very recent results [5], [8] that will guarantee that the discrete strategy of the pursuer will be feasibly executed by the dynamic model of the pursuer. Under suitable

conditions, a feedback controller is constructed for each triangle, steering the pursuer between adjacent triangles. The randomized strategy between triangles will result in a hybrid controller for the pursuer, where the switching among the triangle-dependent controllers will be orchestrated by the randomized pursuit-evasion strategy. The combination of the two results gives rise to a stochastic hybrid controller for the pursuer that can capture the evader with probability arbitrarily close to one. This is one of the few results in the literature for pursuit-evasion games in complex environments which give global guarantees while ensuring that the generated motions are feasible.

The paper is organized as follows: In Section II, we present a formal definition of the visibility based pursuit-evasion game. An overview of the discrete strategy based on the triangulation graph of the environment is presented in Section III. Next, in Section IV, we show how control inputs for implementing the discrete pursuit strategy can be generated. We conclude the paper with simulations (Section V) and an overview of our results (Section VI).

II. PROBLEM FORMULATION

A. Environment Description

Let P be the polygon that represents the environment where the game is played. Throughout the paper, we will use P to denote both the boundary and the interior of the environment. Unless stated otherwise, n denotes the number of vertices of P . We say two points $x, y \in P$ can *see* each other if the line segment xy lies entirely in P .

A polygon is *simply-connected* if it contains no holes, i.e. any simple closed curve inside the polygon can be shrunk to a point. All the polygons considered in this paper are simply-connected. The *triangulation* of a polygon is a decomposition of the polygon into triangles by a maximal set of non-intersecting diagonals. The dual of a triangulation is a graph whose vertices correspond to the triangles. There is an edge between two vertices if the corresponding triangles share a side. See Figure 2 for an illustration. It is well known that the triangulation of a simply-connected polygon has exactly $n-2$ triangles where n is the number of vertices of the polygon. In addition, the dual of the triangulation of a simply-connected polygon contains no cycles, it is a tree [17].

B. Game Formulation

In this section, we formally define the visibility-based pursuit evasion game. There are two players,

a pursuer and an evader. The motion of the pursuer is subject to the following planar fully-actuated kinematics:

$$\dot{x} = u, x \in P, u \in U \quad (1)$$

where $x \in P \subset \mathbb{R}^2$, and the control u is bounded to a polyhedral subset U of \mathbb{R}^2 . Here, P denotes the polygon where the game is played. In this paper, we present results for single integrators on the plane. In the full version of the paper, we present generalizations to more complicated, even nonlinear, dynamics [5].

In this game, the evader is much more powerful than the pursuer. In fact it can be modeled as the pursuer above, but with no constraints on u . It can thus be arbitrarily faster than the pursuer. Furthermore, it knows the position of the pursuer at all times.

The game takes place in a simply-connected polygon P . The pursuer's initial position is an arbitrary point inside P and is known to the evader. However, the pursuer does not know the initial position of the evader.

When the game starts, the pursuer starts searching for the evader. The pursuer wins the game if in finite time it can see, or locate the evader. The evader wins the game if it can avoid being seen forever.

It is worth mentioning that we make no assumptions about the strategy of the evader who actively avoids being seen. As mentioned before, the evader knows the position of the pursuer at all times and it can adaptively design a strategy based on the current position of the pursuer.

The question is then, can we design a pursuer strategy so that the evader will eventually be located no matter which strategy it follows?

III. THE PURSUER STRATEGY FOR LOCATING THE EVADER

In [7] it has been shown that there are simply-connected environments such that if the pursuer is restricted to deterministic strategies, there are polygons with n vertices such that $O(\log n)$ pursuers are required to locate the evader. However, a single pursuer can locate the evader in *any* simply-connected environment with probability arbitrarily close to one – using a randomized strategy [10]. In this section, we give an overview of the randomized pursuit-strategy for a robot with no motion constraints.

Given polygon P , the pursuer first triangulates the polygon. Let $d(u, v)$ denote the minimum travel time from vertex u to vertex v . We define $diam(P)$ as the

maximum amount of time it takes to travel between any two vertices of the polygon, i.e. $\text{diam}(P) = \max_{u,v \in P} d(u,v)$. The pursuer's strategy is divided into rounds of length at most $\text{diam}(P)$. Let T be the dual triangulation tree rooted at the triangle that contains the pursuer's initial location at the beginning of a round. For any triangle t let t_1, \dots, t_k , $k \leq 3$ be the children of t . We use the notation $T(t)$ to denote the subtree of T rooted at the triangle t . Figure 1 is provided for quick reference to the notation used in this section.

The pursuer's strategy relies on the following observation: Suppose the pursuer is inside triangle t and the evader is located inside a triangle contained in $T(t_j)$ for some j . Then, while the pursuer is located at t , the evader can not enter any triangle contained in $T(t_i)$, $i \neq j$ without being seen by the pursuer. This is because the triangle t is a separator for the subtrees $T(t_i)$. Moreover, this property is preserved if the pursuer moves to the triangle t_j .

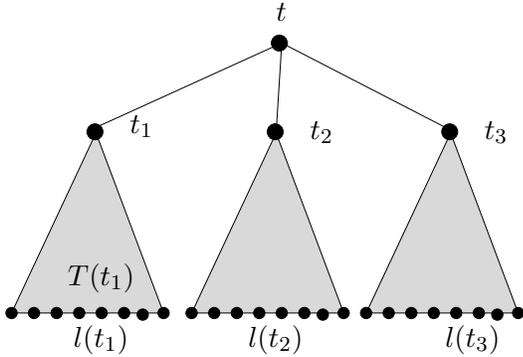


Fig. 1. Each vertex of the tree corresponds to a triangle in the triangulation tree. When located at t , the pursuer picks triangle t_1 with probability $l(t_1)/(l(t_1) + l(t_2) + l(t_3))$ where $l(t_i)$ denotes the number of leaves of the subtree rooted at t_i .

Therefore, had the pursuer known the subtree that contains the evader, he could gradually move towards it while preventing the evader to move from one subtree to another. This process guarantees that the pursuer can enter the triangle the evader is located in and this clearly implies that the evader would be located. Of course, the pursuer does not know where the evader is. This is where we will utilize randomization. The pursuer will guess the subtree that contains the evader according to the following rule:

Let $l(t)$ denote the number of leaves of the subtree $T(t)$. Suppose the pursuer is located in triangle t and let t_1, \dots, t_k be the children of t (see Figure 1). Let $L = \sum_{i=1}^k l(t_i)$. With probability $\frac{l(t_i)}{L}$, the pursuer picks the child t_i and moves there. After arriving at t_i , the pursuer randomly picks one of the children of t_i using the same weighted guessing strategy. The

round is over whenever the pursuer arrives at a leaf of T .

It has been shown [10] that the probability of finding the evader in each round is at least $\frac{1}{n}$. Therefore, if the pursuer repeats this strategy for $n \log n$ rounds, the probability of capturing the evader will be at least $1 - \frac{1}{n}$. This probability can be made arbitrarily close to 1 by increasing the number of rounds.

The following theorem summarizes this result.

Theorem 1 ([10]): In any simply connected environment P , against any evader strategy, the expected time to locate the evader with a single pursuer is at most $n \cdot \text{diam}(P)$ where n is the number of vertices and $\text{diam}(P)$ is the diameter of the polygon.

The high-level strategy for finding the evader is presented in Table I.

LocateTheEvader(T: a triangulation of the environment)
while the evader is not found
$t \leftarrow$ current triangle of the pursuer
$T \leftarrow T$ rooted at t
repeat
$C \leftarrow \{t_i : t_i \text{ is a child of } t\}$
$t_{next} \leftarrow$ randomly chosen triangle from C where
t_i is chosen with probability $\frac{l(t_i)}{\sum_j l(t_j)}$
move from t to t_{next}
$t \leftarrow t_{next}$
until t is a leaf triangle

TABLE I

THE PURSUER'S STRATEGY FOR LOCATING THE EVADER. THE NOTATION $l(t)$ DENOTES THE NUMBER OF LEAVES OF THE SUBTREE ROOTED AT t (WITH RESPECT TO THE TRIANGULATION TREE T , SEE FIGURE 1).

To be able to implement the algorithm presented in Table I, we must generate control inputs that take the pursuer from the current triangle t to the next triangle t_{next} . We address this problem in the next section.

IV. MOTION PLANNING IN TRIANGULATED ENVIRONMENTS

The goal of this section is to plan motions for the pursuers that on one hand implement the discrete strategy of moving from one triangle to another, but on the other are compatible with the pursuer dynamics and input constraints.

Consider triangle S_2 from the triangulation of P , the input polygon.¹ Consider three affinely independent points v_1, v_2, v_3 in $P \subset \mathbb{R}^2$. The triangle S_2 with vertices v_1, v_2, v_3 can be expressed as the convex hull of v_1, v_2, v_3 :

¹We use the notation S_2 to emphasize the fact that the triangles are simplices in \mathbb{R}^2 .

$$S_2 = \{x \in \mathbb{R}^2 \mid x = \sum_{i=1}^3 \lambda_i v_i, \sum_{i=1}^3 \lambda_i = 1, \lambda_i \geq 0\} \quad (2)$$

For $i \in \{1, 2, 3\}$, the convex hull of $\{v_1, v_2, v_3\} \setminus \{v_i\}$ is a facet of S_2 and is denoted by F_i . Let n_i denote the corresponding unit outer normal vector. Consider the following control system

$$\dot{x} = u, \quad x \in S_2 \quad (3)$$

where the control u is bounded to a polyhedral subset U of \mathbb{R}^2 . We are interested in determining constrained linear feedback control laws

$$u = k(x) = Ax + b \in U, \quad (4)$$

where $A \in \mathbb{R}^{2 \times 2}$ and $b \in \mathbb{R}^2$, with the property that all the initial states in S_2 are driven out of S_2 through a desired facet in finite time.

The solution to this problem has been recently shown in [8], [5] for the general case of an n -dimensional simplex.

Lemma 2: The affine function (4) is uniquely determined by its values $k(v_i) = g_i$, $i = 1, 2, 3$ at the vertices of S_2 . Moreover, the restriction of k to S_2 is a convex combination of its values at the vertices and is given by:

$$k(x) = GV^{-1} \begin{bmatrix} x \\ 1 \end{bmatrix}, \quad x \in S_2 \quad (5)$$

where

$$G = [\quad g_1 \quad g_2 \quad g_3 \quad] \quad (6)$$

and

$$V = \begin{bmatrix} v_1 & v_2 & v_3 \\ 1 & 1 & 1 \end{bmatrix} \quad (7)$$

are 2×3 and 3×3 real matrices.

Remark 1: The restriction of an affine function k to a facet F_i of S_2 (F_i itself is a "triangle" in \mathbb{R}^1 , i.e., a line segment) is affine and for any $x \in F_i$, $k(x)$ is a convex combination of the values of k at the vertices of F_i .

Proposition 3: There exists an affine feedback law (4) driving all initial states in the simplex S_2 through the facet F_i in finite time if and only if the following sets are nonempty:

$$U_i = U \cap \{g \in \mathbb{R}^2 \mid n_j^T g \leq 0, \\ j = 1, 2, 3, j \neq i \text{ and } n_i^T g > 0\},$$

$$U_j = U \cap \{g \in \mathbb{R}^2 \mid n_i^T g > 0 \text{ and } \\ n_k^T g \leq 0 \text{ for all } k = 1, 2, 3, k \neq j, k \neq i\}$$

for all $j = 1, 2, 3$, $j \neq i$.

If one of the sets from Proposition 3 is empty, then there is no affine feedback law in S_2 satisfying the corresponding property. If they are all nonempty, then any choice of $g_i \in U_i$, $i = 1, 2, 3$ will give a valid affine vector field by formula (5). Indeed, for every $x \in S_2$, we know that $k(x)$ is a convex combination of $g_1, g_2, g_3 \in U$. Hence, $k(x)$ is contained in the convex hull of g_1, g_2, g_3 , which is the smallest convex set containing g_1, g_2, g_3 , and therefore included in U . So the vector field is bounded everywhere in the simplex as required, and achieves the desired goal of steering the pursuer from one triangle to another (adjacent) triangle.

Using Proposition 3 in each of the triangular regions, we can derive necessary and sufficient conditions for the existence of affine vector fields (restricted to the polyhedral set U) driving all initial states through a separating facet in finite time.

Note that, if we choose the same velocity values at the vertices corresponding to the common facet of two adjacent triangles, the continuity of the vector field is guaranteed everywhere. Indeed, the vector fields in two adjacent triangles coincide on the separating facet, since their restrictions to the separating facet, which is a lower dimensional simplex, are uniquely determined by the values at the corresponding vertices. Sample trajectories for adjacent triangles are shown in Figure 2.

Remark 2: Integrating the results of the previous two sections can be naturally captured in the language of hybrid systems [3]. The triangulation of the environment P results in a finite partition of the state space. In every element (triangle) of the partition, the pursuer is evolving using dynamics given in Equation 3 under the influence of affine, feedback controller of Equation 4 that guarantee physical motion between adjacent triangles. The switching occurs when the pursuer reaches the facet of adjacent triangles. The probabilities of the discrete transitions obey the guessing rules obtained in Section III.

V. SIMULATIONS

In Figure 2, we present sample trajectories generated for a point robot (the pursuer). The velocity of the pursuer is subject to the polyhedral bounds $U = [-1, 1] \times [0, 1]$. The environment where the game takes place, its triangulation and the dual graph of the triangulation is shown in the top figure. Actual trajectories generated for four different rounds of the game are presented in the rest of the figures. Each round corresponds to a trip from one leaf to

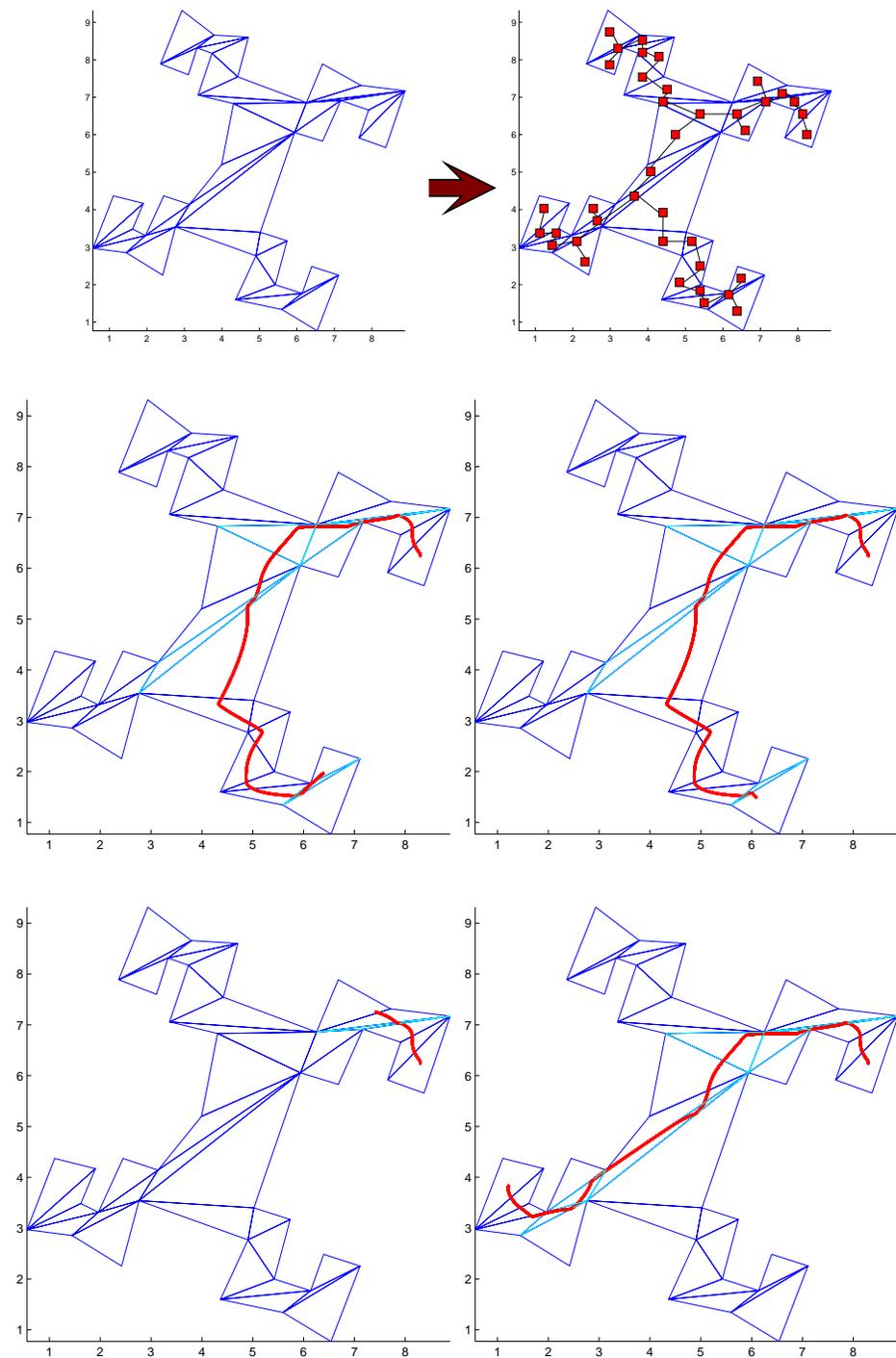


Fig. 2. **TOP:** A polygonal environment, its triangulation and the dual of the triangulation **MIDDLE-BOTTOM ROWS:** Actual trajectories generated for four different rounds of the game. Each round corresponds to a trip from one leaf to another. Even though the rounds start with identical initial conditions, due to the randomized nature of the strategy, different trajectories are generated

another. During the round, the pursuer picks one of the children of his current triangle randomly as described in Section III.

VI. CONCLUSION

In this paper, we have studied the problem of generating feasible trajectories for a pursuer who tries to locate an adversarial, unpredictable evader in a simply-connected polygon. Our approach starts from a discrete, randomized pursuit strategy based on a triangulation of the environment. We then generate feasible trajectories that obey motion constraints of the pursuer's model. The overall strategy yields a stochastic hybrid controller for the pursuer which guarantees that, with probability arbitrarily close to one, the pursuer will locate the evader regardless of its strategy. This is one of the few results in the literature for pursuit-evasion games in complex environments which give global guarantees while ensuring that the generated motions are feasible.

One of our future research directions is to study the visibility-based pursuit evasion game in multiply-connected environments. Note that since the evader knows the position of the pursuer at all times, we can not locate the evader in such an environment using a single pursuer. Other research directions include pursuit-evasion games in higher dimensions as well as more ambitious notions of capture such as intercepting the evader.

ACKNOWLEDGMENT

Research performed at the University of Pennsylvania is partially supported by the Army Research Office under MURI Grant DAAD 19-02-01-0383.

REFERENCES

[1] M. Adler, H. Racke, N. Sivadasan, C. Sohler, and B. Vocking. Randomized pursuit-evasion in graphs. *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, 2002.

[2] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Math*, 8:1–12, 1984.

[3] R. Alur, T. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(2):971–984, jul 2000.

[4] T. Basar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. SIAM, 1998.

[5] C. Belta and L. Habets. Constructing decidable hybrid systems with velocity bounds. In *43rd IEEE Conference on Decision and Control*, 2004. Submitted.

[6] G. Brightwell and P. Winkler. Gibbs measures and dismantlable graphs. *J. Comb. Theory (Series B)*, 78, 2000.

[7] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani. A visibility-based pursuit-evasion problem. *International Journal of Computational Geometry and Applications*, 9(4/5):471–, 1999.

[8] L. Habets and J. van Schuppen. A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica*, 40:21–35, 2004.

[9] R. Isaacs. *Differential Games*. Dover, 1965.

[10] V. Isler, S. Kannan, and S. Khanna. Locating and capturing an evader in a polygonal environment. In *Workshop on Algorithmic Foundations of Robotics (WAFR'04)*, 2004. to appear.

[11] V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion with limited visibility. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1053–1063, 2004.

[12] S. M. LaValle and J. Hinrichsen. Visibility-based pursuit-evasion: The case of curved environments. *IEEE Transactions on Robotics and Automation*, 17(2):196–201, Apr. 2001.

[13] J.-H. Lee, S.-M. Park, and K.-Y. Chwa. Searching a polygonal room with one door by a 1-searcher. *International Journal of Computational Geometry and Applications*, 10(2):201–220, 2000.

[14] J. Lewin. *Differential Games*. Springer Verlag, 1994.

[15] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou. The complexity of searching a graph. *J. ACM*, 1988.

[16] R. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Math*, 43:235–239, 1983.

[17] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1998.

[18] S.-M. Park, J.-H. Lee, and K.-Y. Chwa. Visibility-based pursuit-evasion in a polygonal region by a searcher. *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, 2076, 2001.

[19] S. Rajko and S. M. LaValle. A pursuit-evasion bug algorithm. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 1954–1960, 2001.

[20] J. Sgall. Solution of David Gale's lion and man problem. *Theoret. Comput. Sci.*, 259(1-2):663–670, 2001.

[21] I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal on Computing*, 21(5):863–888, 1992.

[22] R. Vidal, O. Shakernia, J. Kim, D. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation. *IEEE Transactions on Robotics and Automation*, 18:662–669, 2002.