Surveillance for Security as a Pursuit-Evasion Game

Sourabh Bhattacharya[1], Tamer Başar[2], and Maurizio Falcone[3]

[1] Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA
sbhattac@iastate.edu
[2] Department of Electrical and Computer Engineering, University of Illinois at Urbana Champaign, Urbana, IL 61801, USA
basar1@illinois.edu
[3] Dipartimento di Matematica, Universita di Roma "La Sapienza", p. Aldo Moro 2, 00185 Roma, Italy
falcone@mat.uniroma1.it

Abstract. This work addresses a visibility-based target tracking problem that arises in autonomous surveillance for covert security applications. Consider a mobile observer, equipped with a camera, tracking a target in an environment containing obstacles. The interaction between the target and the observer is assumed to be adversarial in order to obtain control strategies for the observer that guarantee some tracking performance. Due to the presence of obstacles, this problem is formulated as a game with state constraints. Based on our previous work in [6] which shows the existence of a value function, we present an off-line solution to the problem of computing the value function using a Fast Marching Semi-Lagrangian numerical scheme, originally presented in [15]. Then we obtain the optimal trajectories for both players, and compare the performance of the current scheme with the Fully Discrete Semi-Lagrangian Scheme presented in [6] based on simulation results.

Keywords: pursuit-evasion games, semi-Lagrangian schemes, fast marching.

1 Introduction

Security is an important concern in infrastructure systems. Although advanced electronic and biometric techniques can be used to secure facilities reserved for military activities, vision-based monitoring is primarily used for persistent surveillance in buildings accessible to civilians. The idea is to cover the environment with cameras in order to obtain sufficient visual information so that appropriate measures can be taken to secure the area in case of any suspicious activity. However, the number of static cameras needed to cover and monitor activities in a moderately sized building is substantial, and this leads to fatigue in security personnel. In this work, we explore a scenario in which mobile agents that can visually track entities in the environment are deployed in a surreptitious manner for surveillance applications. This gives rise to a problem that is often called the *target tracking* problem.

R. Poovendran and W. Saad (Eds.): GameSec 2014, LNCS 8840, pp. 370–379, 2014.

Target tracking refers to the problem of tracking a mobile object, called a *target*. Based on the sensing modality and sensing constraints, there is a range of problems that can be addressed under this category. In this work, we assume that the autonomous observer is equipped with a vision sensor for tracking the target. The environment contains obstacles that occlude the view of the target from the observer. The goal of the observer is to maintain a persistent line-of-sight with the target. Therefore, the mobile observer has to control its motion, keeping in mind the sensing constraints and the motion constraints posed by the obstacles. In order to compute motion strategies for the observer that can provide some performance guarantees, the target is assumed to be an adversary. Several variants of the target-tracking problem have been considered in the past that consider constraints in motion as well as sensing constraints for both agents. For an extensive discussion regarding the previous work and its applications, we refer to [13,12]. In this work, we consider the target tracking problem without any constraints in sensing or motion models for both agents except for those posed by the obstacles present in the environment.

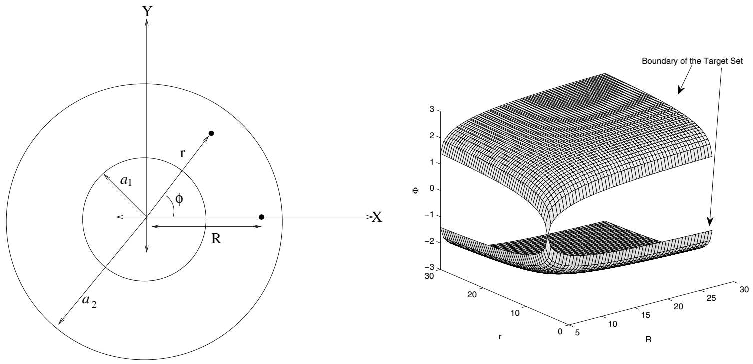
Past efforts to provide a solution to the aforementioned problem can be primarily divided into two categories: (1) Formulating the problem as a game of kind, and providing necessary conditions for pursuit and evasion in the presence of polygonal obstacles [13,10,11]; (2) Formulating the problem as a game of degree, and using the theory of differential games to provide necessary and sufficient conditions for pursuit [12,14,7,9]. Although, the structure of optimal solutions has been characterized extensively in previous works, a complete construction of the solution in a general environment containing polygonal obstacles is still open. In [8], the authors analyze the problem in a simple environment containing a circular obstacle, and characterize the optimal trajectories near termination using differential game theory. In [6], we use a semi-Lagrangian iterative numerical scheme to provide a solution to the aforementioned problem. In this work, we use another numerical technique, *Fast Marching Semi-Lagrangian scheme*, based on the ideas of front propagation to provide an off-line solution to the problem. The numerical techniques introduced in this work can be used for any 2-player generalized pursuit-evasion game with state constraints.

Numerical techniques for games are primarily based on the principles of Dynamic Programming (DP). Finite differences approximation schemes based on generalized gradients were proposed by Tarasyev[20] who also considered the problem of the synthesis of optimal controls using approximate values on the finite grid. Convergence results to the value function of the generalized pursuit-evasion games for the approximation scheme based on Discrete Dynamic Programming (also called semi-Lagrangian scheme) were first presented in [4], under either continuity assumptions on the value function or for problems with a single player (i.e. control problems). The extension of the scheme and of the convergence theorem to the discontinuous case was obtained in [2]. Later these results have been extended to pursuit-evasion games with state constraints in [5,16]. Our work is in a similar vein, and uses the fully discrete scheme proposed in the aforementioned works to address the target tracking game. For a general introduction to semi-Lagrangian schemes and their applications in control and game problems, we refer to [18].

The paper is organized as follows. In Section 2, we present the problem formulation, and address the issue of existence of the value function for our problem setting. In Section 3, we reduce the dimensionality of the problem by reformulating it in relative coordinates. In Section 4, we present the numerical scheme. In Section 5, a comparison of the different schemes is presented based on simulation results. Finally, Section 6 includes some concluding remarks.

2 Problem Statement

In this section, we present the problem formulation (see Figure 1(a)). Consider a circular obstacle in the shape of a disc of radius a_1 in the plane enclosed inside a concentric circular boundary of radius a_2. The centers of both circles are assumed to be at the origin of the reference frame. Consider a mobile observer and a target in the plane. Each agent is assumed to be a point in the plane. Let $\mathbf{y} \in \mathbb{R}^2$ and $\mathbf{z} \in \mathbb{R}^2$ denote the coordinates of, respectively, the observer and the target in the plane. Both agents are assumed to be



(a) **(b)**

Fig. 1. Figure (a) shows the geometry around a circular obstacle with a circular boundary. Figure (b) shows the boundary of the terminal manifold of the game in relative coordinates.

simple kinematic agents, and their motions are governed by the following equations

$$\dot{\mathbf{y}} = \mathbf{u}_1, \ \dot{\mathbf{z}} = \mathbf{u}_2$$

subject to the constraints $\mathbf{y} \in K_U$, $\mathbf{z} \in K_V$ where

$$K_U \equiv \{\mathbf{y} \in \mathbb{R}^2 : (\|\mathbf{y}\|_2^2 - a_1^2)(\|\mathbf{y}\|_2^2 - a_2^2) \leq 0\}, \ K_V \equiv \{\mathbf{z} \in \mathbb{R}^2 : (\|\mathbf{z}\|_2^2 - a_1^2)(\|\mathbf{z}\|_2^2 - a_2^2) \leq 0\}$$

Let $\mathbf{x} = (\mathbf{y}, \mathbf{z})^T$ and $f(\mathbf{x}, \mathbf{u}_1, \mathbf{u}_2) = (\mathbf{u}_1, \mathbf{u}_2)^T$. The controls $\mathbf{u}_1(\cdot)$ and $\mathbf{u}_2(\cdot)$ belong to the following sets

$$\mathbf{u}_1(\cdot) : \mathbb{R} \to U, \quad U = B_1(0,0), \quad \mathbf{u}_2(\cdot) : \mathbb{R} \to V, \quad V = B_\mu(0,0)$$

where $B_r(a)$ is a ball of radius r with center a, and μ is a parameter which represents the maximum speed of the target. We will see later that we have to pick $\mu \leq 1$ to make the problem meaningful. The line-of-sight between the pursuer and the evader is defined as the line joining the two players on the plane. The line-of-sight is considered to be broken if it intersects with the circular obstacle. In order to account for the worst-case scenario, the target is assumed to be adversarial in nature. Therefore, the interaction between the observer and the target is modeled as a game. The observer is assumed to be the pursuer, and the target is assumed to be the evader. The objective of the pursuer is to maximize the time for which it can continuously maintain a line of sight to the evader. The objective of the evader is to break the line-of-sight in the minimum amount of time. The game terminates when the line-of-sight between the pursuer and the evader is broken. The problem is to compute the strategies of the players as a function of their positions. Since this is a 2-player zero-sum game [1], we use the concept of *saddle-point equilibrium* [12] to define the optimal strategy for each player.

Let $T(\mathbf{x_0})$ denote the optimal time of termination of the game when the players start from the initial position $\mathbf{x_0}$. A strategy for a player will be defined as a map from the control set of the opponent to its own control set, with some informational constraints imposed, as appropriate. Let α and β denote the strategies of the pursuer and the evader, respectively. A pair of strategies (α^*, β^*) for the two players is said to be in saddle-point equilibrium if the following pair of inequalities is satisfied

$$T(\mathbf{x_0}; \alpha^*, \beta) \geq T(\mathbf{x_0}; \alpha^*, \beta^*) \geq T(\mathbf{x_0}; \alpha, \beta^*) \qquad \forall \alpha, \beta \text{ admissible}$$

(here we write explicitly the dependence of T on the strategies). If the pair (α^*, β^*) exists, then the function $T^*(\mathbf{x_0}) = T(\mathbf{x_0}; \alpha^*, \beta^*)$ is called the value of the game and T^* is called the *value function*. The existence of the value function depends on the class of strategies under consideration for both the players. In this work, the notion of *non-anticipating strategies* [17] will be used to define the information pattern between the players.

Definition: A strategy α for player P is non-anticipating if $\alpha \in \Gamma$, where

$$\Gamma = \{\alpha : V \to U \,|\, b(t) = \tilde{b}(t),\ \forall t \leq t' \text{ and } b(t), \tilde{b}(t) \in V \Rightarrow \alpha[b](t) = \alpha[\tilde{b}](t),\ \forall t \leq t'\}$$

Similarly, we can define a non-anticipating strategy $\beta \in \Delta$ for E, where

$$\Delta = \{\beta : U \to V \,|\, a(t) = \tilde{a}(t),\ \forall t \leq t' \text{ and } a(t), \tilde{a}(t) \in V \Rightarrow \beta[a](t) = \beta[\tilde{a}](t), \forall t \leq t'\}$$

Frequently, in problems involving games and optimal control, it is the case that the value function ceases to exist in the class of strategies used by the players. In [6], we show that the value of the game exists. Since the existence of the value function is established from the above transversality conditions, we can address the problem of computing it.

3 Dimensionality Reduction

In this section, we present a formulation of the problem in reduced coordinates where we exploit the symmetry of the problem in order to reduce dimensionality. To this end, we formulate the problem in polar coordinates. We express the position of the players in relative coordinates. Let the polar coordinates of the pursuer and the evader be denoted as (r_p, θ_p) and (r_e, θ_e), respectively. Instead, we can use the relative coordinates ($R = r_p, r = r_e, \phi = (\theta_p - \theta_e)$) to define the state of the game. The equations of motion of the two players in relative coordinates are given by the following

$$f_R = \dot{R} = u_{r_p}; \quad f_r = \dot{r} = u_{r_e}; \quad f_\phi = \dot{\phi} = \frac{u_{\theta_e}}{r} - \frac{u_{\theta_p}}{R}, \tag{1}$$

where (u_{r_p}, u_{θ_p}) and (u_{r_e}, u_{θ_e}) are the radial and tangential components of the velocities of the pursuer and the evader, respectively, and satisfy the following constraints

$$u_{r_p}^2 + u_{\theta_p}^2 \leq 1; \quad u_{r_e}^2 + u_{\theta_e}^2 \leq \mu^2 \tag{2}$$

The problem statement dictates that $a_1 \leq R, r \leq a_2$ and $-\pi \leq \phi \leq \pi$. The problem is to determine the time of termination of the game, and the optimal strategies of the individual players given the initial position $\mathbf{x} = (r, R, \phi)$ of the pursuer and the evader:

$$(u_{r_p}^*, u_{\theta_p}^*, u_{r_e}^*, u_{\theta_e}^*) = \arg \max_{u_{r_p}, u_{\theta_p}} \min_{u_{r_e}, u_{\theta_e}} T(\mathbf{x}; u_{r_p}, u_{\theta_p}, u_{r_e}, u_{\theta_e}) \tag{3}$$

The existence of the value function was established in [6], as indicated in the previous section, and hence the max and min operations commute in the above equation. Since the evader always wins from any given initial position of the players for $\mu > 1$, we only consider the case $\mu \leq 1$. The winning strategy of the evader for $\mu > 1$ is to move along the boundary of the obstacle with its maximum speed in a fixed direction. Based on the problem formulation, the game terminates when the line-of-sight between the pursuer and the evader intersects with the circular obstacle. Therefore, the boundary of the terminal manifold is given by the set of states for which the line-of-sight between the pursuer and the evader is tangent to the circular obstacle.

Figure 1(b) shows the boundary of the terminal manifold in relative coordinates for $a_1 = 5$ and $a_2 = 30$. The line-of-sight is in the free space only if the state of the players lies between the two symmetric surfaces. Otherwise, the game has terminated. The set of states for which the line-of-sight intersects the obstacles is also the target set, denoted as $\mathscr{T}$. The objective of the evader is to drive the state of the system to the target set. The objective of the pursuer is to prevent the state from reaching it. Let $\mathscr{R}$ denote the reachable set, i.e., the set of initial points from which it is possible for the evader to drive the state of the system to the target set in finite time irrespective of the pursuer's control action. One can clearly see that $\mathscr{R}$ depends on $\mathscr{T}$ and the dynamics of the players.

We have the following result from [4].

Theorem 1. *If $\mathscr{R} \setminus \mathscr{T}$ is open, and $T \in C^0(\mathscr{R} \setminus \mathscr{T})$, then $T(\cdot)$ is a viscosity solution of the following equation:*

$$\min_{\mathbf{a} \in U} \max_{\mathbf{b} \in V} \{-f(\mathbf{x}, \mathbf{a}, \mathbf{b}) \cdot \nabla T(\mathbf{x})\} - 1 = 0, \quad \mathbf{x} \in \mathscr{R} \setminus \mathscr{T} \tag{4}$$

Let $\mathrm{v}(\mathbf{x})$ denote the *Kružkov transform* [3] of $T(\mathbf{x})$

$$\mathrm{v}(\mathbf{x}) = \begin{cases} 1 - e^{-T(\mathbf{x})} & \text{if } T(\mathbf{x}) < +\infty \quad (\mathbf{x} \in \mathscr{R}) \\ 1 & \text{if } T(\mathbf{x}) = +\infty \quad (\mathbf{x} \notin \mathscr{R}) \end{cases} \tag{5}$$

Since $T(\mathbf{x})$ takes values in the interval $[0,\infty)$, $\mathrm{v}(\mathbf{x})$ takes values in the interval $[0,1]$. Using $\mathrm{v}(\mathbf{x})$ instead of $T(\mathbf{x})$ leads to better numerical schemes due to the bounded values of $\mathrm{v}(\mathbf{x})$. Moreover, there is a bijective map between $\mathrm{v}(\mathbf{x})$ and $T(\mathbf{x})$ given by the following:

$$T(\mathbf{x}) = -ln(1-\mathrm{v}(\mathbf{x}))$$

In terms of $\mathrm{v}(\mathbf{x})$, the reachable set is given by the following expression

$$\mathscr{R} = \{\mathbf{x} | \mathrm{v}(\mathbf{x}) < 1\}$$

Therefore, we address the problem of computing $\mathrm{v}(\mathbf{x})$ numerically in the following sections. If $\mathrm{v}(\mathbf{x})$ is continuous, then it is the unique viscosity solution of the following Dirichlet problem [4]

$$\begin{cases} \mathrm{v}(\mathbf{x}) + \min_{\mathbf{a}\in U} \max_{\mathbf{b}\in V} \{-f(\mathbf{x},\mathbf{a},\mathbf{b}) \cdot \nabla \mathrm{v}(\mathbf{x})\} - 1 = 0, & \text{for } \mathbf{x} \in \mathbb{R}^n \setminus \mathscr{T} \\ \mathrm{v}(\mathbf{x}) = 0 & \text{for } \mathbf{x} \in \partial\mathscr{T} \end{cases}$$

4 Numerical Scheme

First, we describe the discretization of the state space. The entire state space $\mathbf{X}(\mathbb{R}^3)$ is discretized by constructing a three dimensional lattice of cubes with edge lengths k. The lattice points are placed at the corners of cubes with the origin as one of the lattice points. The numerical scheme computes the approximation of $\mathrm{v}(\mathbf{x})$ at the lattice points. Let Q denote a closed and bounded subset of $\mathbf{X}$ containing the entire free space including the obstacles. Once the state space is discretized, we are only concerned with values of v at those lattice points which belong to Q. We will call these lattice points as *nodes*. Let the nodes be ordered as $\{1,\ldots,N\}$, where N is the number of nodes in Q. Let $(\mathbf{x}_1,\ldots,\mathbf{x}_N)$ denote the state of the nodes in Q. Let $I_{\mathscr{T}}$ denote the set of nodes in Q that belong to the target set. The values of these nodes are set to zero since the game would already have terminated if it started from any of these nodes. Therefore, if $x_i \in I_{\mathscr{T}}$, $T_{\mathbf{x_i}} = 0$, which implies $\mathrm{v}(\mathbf{x_i}) = 0$. We arrange the values of v at all the nodes in the form of a vector $V = (V_1,\ldots,V_N)$. The solution is usually obtained via a fixed point iteration $V^{n+1} = SV^n$ starting from a given V^0 [18].

In the Fast Marching Method (FMM), the state space is initially discretized in a manner described in the previous paragraph. At every instant of time, the nodes are divided into the following three groups. The *accepted nodes* are those where the solution has already been computed, and it cannot change in the subsequent iterations. The *narrow band nodes* are those where the computation actually takes place, and their values can change in the subsequent iterations. The *far nodes* are those in the space where an approximate solution has never been computed. The front in our problem represents the surface that updates the initial value of $\mathrm{v}(\mathbf{x}_i)$ at node i to its approximate value as it propagates in the state space. The accepted region represents the nodes in the state space through which the front has already passed. The narrow band represents the nodes in the region around the current position of the front where the values are being updated. The far region represents the nodes where the front has not yet passed.

The algorithm initializes by labeling all the nodes in the target set as accepted nodes. In order to compute the narrow band nodes, we need to first define the concept of *reachable sets*. The reachable set at any iteration is defined as the set of nodes from which the pursuer can drive the state of the system to a node that belongs to the accepted set irrespective of the controls of the evader. A sketch of the algorithm is given below:

1. The nodes belonging to the target set $\mathscr{T}$ are located and labeled as *accepted*, setting their values to $\mathrm{v}(x) = 0$. All other nodes are set to $\mathrm{v}(x) = 1$ and labeled as *far*.
2. The initial *narrow band* is defined as the set of all the neighbors of the accepted nodes. Their values are valid only if they are in the reachable set.
3. The node in the narrow band with the minimal valid value is accepted, and it is removed from the narrow band.
4. Neighbors of the last accepted node that are not yet accepted are computed and inserted in the narrow band. Their values are valid only if they are in the reachable set.
5. If the narrow band is not empty, the next iteration starts at step 3.

The complete algorithm is given in the table below as Algorithm 1.

Algorithm 1. FMSL

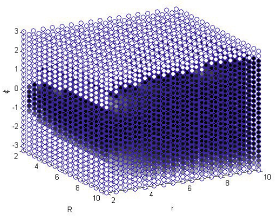
declare $\mathscr{Q}_{Accepted}, \mathscr{Q}_{NarrowBand}, \mathscr{Q}_{Far}$ be the sets of accepted nodes, narrow band nodes and far nodes
for each $x_i \in \mathscr{Q}$ **do**
    if $x_i \in \mathscr{T}$ **then**
        $V_{x_i} = 0$ and $x_i \in \mathscr{Q}_{Accepted}$
    else
        $V_{x_i} = 1$ and $x_i \in \mathscr{Q}_{Far}$
    end if
end for
$\mathscr{Q}_{NarrowBand} = \{x_j | x_j \in \bigcup_{x_i \in \mathscr{Q}_{Accepted}} N(x_i) \cap \mathscr{R}^h\}$
while $\mathscr{T}_{NarrowBand} \neq \emptyset$ **do**
    if $x_k = \arg\min V(x_j)$ **then**
        Remove x_k from $\mathscr{Q}_{NarrowBand}$ and add it to $\mathscr{Q}_{Accepted}$
        Add $N(x_k) \cap \mathscr{R}^h$ to $\mathscr{Q}_{NarrowBand}$
    end if
end while

From [19], it is well known that the performance of FM deteriorates rapidly when the characteristic and the gradient lines do not coincide. In order to overcome this limitation, the Buffered Fast Marching Method (BFMM) was introduced in [15]. BFMM is an amalgamation of SL and FM methods that retains the advantages of both techniques. In BFMM, in addition to the accepted nodes, narrow band and far nodes, we have a *buffer* zone. Every iteration of BFMM starts with the implementation of the FM scheme. Once the nodes having the least value in narrow band are computed, they are moved to buffer. All the nodes in the buffer are recomputed using the Fully Discrete Semi-Lagrangian scheme for two different initial boundary conditions of the nodes. In the first step, the values of all the nodes in the narrow band are set to 1. In the second step, the values of all the nodes in the narrow band are set to 0. If there is any node in the buffer for which the value remains unchanged with two different boundary conditions, then the node is considered to be accepted.

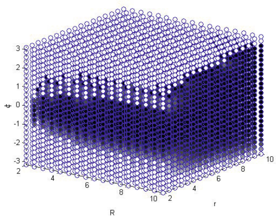
In the next section, we present some numerical results obtained from the FMM and BFMM.

5 Results

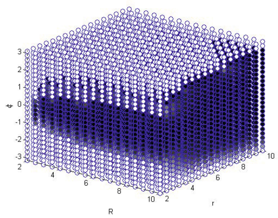
In this section, we present simulation results, and compare it with our previous results in [6]. All the simulations were performed on a Core 2 Duo P7450 processor. The radii of the inner and outer obstacles are $a_1 = 1$ and $a_2 = 10$, respectively. The speed of the evader is set at 0.8 for all simulations. Figure 2 depicts the value function for all the three numerical schemes, and trajectories of the players for a specific initial position. Figure 2(d) shows the trajectories of the players computed from the Fully Discrete Semi-Lagrangian technique presented in [6]. Figures 2(e) and 2(f) show the trajectories of the players from the Fast Marching techniques proposed in this work. Figure 3 illustrates the variation of the performance of the three techniques on the basis of the computational time and capture time with respect to the grid size. Figure 3(a) shows the time expended to compute the value functions for the three different techniques as the grid size increases. We can see that for a fixed grid size the iterative scheme takes more



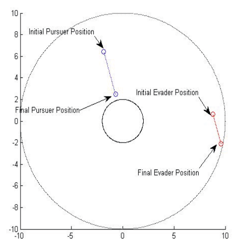
(a) Fully Discrete Semi-Lagrangian Scheme



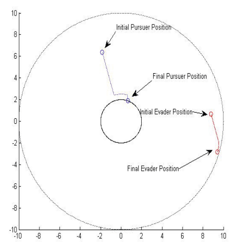
(b) Fast Marching Scheme



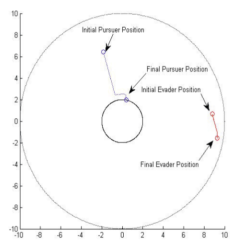
(c) Buffered Fast Marching Scheme



(d) Fully Discrete Semi-Lagrangian Scheme



(e) Fast Marching Scheme



(f) Buffered Fast Marching Scheme

Fig. 2. The figure shows variation of the value function computed at the nodes, and the trajectories of the players for the three techniques

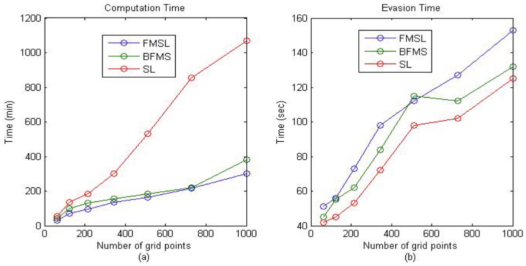


Fig. 3. Figure (a) shows a plot of the computational time required to compute the $v(x_i)$ using the three different techniques. Figure (b) shows the variation of the time required for the target to escape with increasing number of grid points used for computation.

time to compute the value function as compared to the FM schemes. Moreover, the results clearly show that the time required for computation of the value function increases as the grid resolution becomes finer. Figure 3(b) shows the variation of the termination time for the game for a fixed trajectory of the target using the three techniques. One can clearly see that the SL scheme is expensive in terms of computational time compared to the other two techniques.

6 Conclusions

This work has addressed a vision-based surveillance problem for securing an environment. The task of keeping a suspicious target in the observer's field-of-view was modeled as a pursuit-evasion game by assuming that the target is adversarial in nature. Due to the presence of obstacles, this problem was formulated as a game with state constraints. We first showed that the value of the game and the saddle-point strategies of the game exist. Then we obtained the optimal (saddle-point) strategies for the observer from three different numerical techniques based on finite-difference schemes. The relative performance of the three different schemes based on computational time, and degree of approximation was illustrated through simulations.

An immediate extension of this work would be to apply the technique to problems that have non-holonomic agents having more complicated dynamics, for example, a Dubin's car or a differential drive robot. We are also working on extending the current technique to more general environments. A fundamental question that remains open is the existence of the value function and the saddle-point strategies for the game in general polygonal environments.

References

1. Başar, T., Olsder, G.J.: Dynamic Noncooperative Game Theory, 2nd edn. SIAM Series in Classics in Applied Mathematics, Philadelphia (1999)
2. Bardi, M., Bottacin, S., Falcone, M.: Convergence of discrete schemes for discontinuous value functions of pursuit-evasion games. In: New Trends in Dynamic Games and Applications, pp. 273–304. Springer (1995)

3. Bardi, M., Capuzzo-Dolcetta, I.: Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Springer (2008)
4. Bardi, M., Falcone, M., Soravia, P.: Fully discrete schemes for the value function of pursuit-evasion games. Advances in Dynamic Games and Applications 1, 89–105 (1994)
5. Bardi, M., Koike, S., Soravia, P.: Pursuit-evasion games with state constraints: Dynamic programming and discrete-time approximations. Discrete and Continuous Dynamical Systems 6(2), 361–380 (2000)
6. Bhattacharya, S., Başar, T., Falcone, M.: IEEE Conference on Intelligent Robots and Systems (to appear, 2014)
7. Bhattacharya, S., Başar, T., Hovakimyan, N.: Singular surfaces in multi-agent connectivity maintenance games. In: 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 261–266 (2011)
8. Bhattacharya, S., Başar, T., Hovakimyan, N.: Game-theoretic analysis of a visibility based pursuit-evasion game in the presence of a circular obstacle. In: AIP Conference Proceedings, vol. 1479, p. 1222 (2012)
9. Bhattacharya, S., Basar, T., Hovakimyan, N.: On the construction of barrier in a visibility based pursuit evasion game. In: European Control Conference (ECC), pp. 1894–1901. IEEE (2014)
10. Bhattacharya, S., Candido, S., Hutchinson, S.: Motion strategies for surveillance. In: Robotics: Science and Systems (2007)
11. Bhattacharya, S., Hutchinson, S.: Approximation schemes for two-player pursuit evasion games with visibility constraints. In: Robotics: Science and Systems (2008)
12. Bhattacharya, S., Hutchinson, S.: On the existence of Nash equilibrium for a two player pursuit-evasion game with visibility constraints. International Journal of Robotics Research 29(7), 831–839 (2010)
13. Bhattacharya, S., Hutchinson, S.: A cell decomposition approach to visibility-based pursuit evasion among obstacles. International Journal of Robotics Research 30(14), 1709–1727 (2011)
14. Bhattacharya, S., Hutchinson, S., Başar, T.: Game-theoretic analysis of a visibility based pursuit-evasion game in the presence of obstacles. In: Proceedings of American Control Conference, St. Louis, MO, pp. 373–378 (June 2009)
15. Cristiani, E.: A fast marching method for Hamilton-Jacobi equations modeling monotone front propagations. Journal of Scientific Computing 39(2), 189–205 (2009)
16. Cristiani, E., Falcone, M.: Numerical solution of the Isaacs equation for differential games with state constraints. In: 17th IFAC World Congress, vol. 17, pp. 11352–11356 (2008)
17. Elliott, R.J., Kalton, N.J.: The existence of value in differential games Number 1-126. AMS Bookstore (1972)
18. Falcone, M., Ferretti, R.: Semi-Lagrangian Approximation Schemes for Linear and Hamilton Jacobi Equations. SIAM (2014)
19. Sethian, J.A., Vladimirsky, A.: Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms. SIAM Journal on Numerical Analysis 41(1), 325–363 (2003)
20. Taras'yev, A.M.: Approximation schemes for constructing minimax solutions of Hamilton-Jacobi equations. Journal of Applied Mathematics and Mechanics 58(2), 207–221 (1994)