Target Surveillance in Adversarial Environments Using POMDPs

Maxim Egorov and **Mykel J. Kochenderfer**
Department of Aeronautics and Astronautics
Stanford University
Stanford, California 94305
{megorov, mykel}@stanford.edu

Jaak J. Uudmae
Department of Computer Science
Stanford University
Stanford, California 94305
juudmae@stanford.edu

Abstract

This paper introduces an extension of the target surveillance problem in which the surveillance agent is exposed to an adversarial ballistic threat. The problem is formulated as a mixed observability Markov decision process (MOMDP), which is a factored variant of the partially observable Markov decision process, to account for state and dynamic uncertainties. The control policy resulting from solving the MOMDP aims to optimize the frequency of target observations and minimize exposure to the ballistic threat. The adversary's behavior is modeled with a level-k policy, which is used to construct the state transition of the MOMDP. The approach is empirically evaluated against a MOMDP adversary and against a human opponent in a target surveillance computer game. The empirical results demonstrate that, on average, level 3 MOMDP policies outperform lower level reasoning policies as well as human players.

1 Introduction

Adversarial target surveillance arises in many peacekeeping operations in which a strategic target must be monitored in the presence of hostile forces. For example, a voting booth in a politically unstable region may be a target of adversaries and require persistent monitoring. Currently, planning for such missions is done by human experts using a set of guiding principles that balance mission objectives (U.S. Army 2008). However, optimal decision making in hostile environments is difficult due to the potential visual constraints imposed by urban structure and the uncertainty associated with the behavior of adversaries. In this work, we consider the problem of target surveillance in the presence of adversarial ballistic threats. There are two primary challenges associated with this problem: how to accurately model an intelligent adversary and how to plan optimally. An intelligent surveillance planner that solves these two challenges can be used as a high level controller for autonomous systems or as a decision support tool for aiding peacekeeping forces in the field.

One way to capture the stochastic nature of target surveillance is to formulate the problem as a Markov decision process (MDP), or more generally as a partially observable MDP (POMDP) to account for uncertainty in the adversary's location. Problems that include a mixture of fully and partially observable variables in the system state can be formulated as mixed observability MDPs (MOMDPs) to reduce computational complexity (Ong et al. 2009). An example map of an urban environment with a single agent, target, and ballistic threat is shown in Figure 1. The heat map represents the belief, i.e, a probability distribution over the states of the threat, as a representation of uncertainty. In this work, the adversarial surveillance problem was modeled as a MOMDP. The policy resulting from solving the MOMDP considers the dynamics of the surveillance agent, its sensing capabilities, and the behavior model of the threat (the adversary) and optimizes the surveillance strategy based on a predefined reward function. The reward function considers the competing objectives of surveillance persistency and risk mitigation. In this paper, we investigate the feasibility of applying state-of-the-art MOMDP solution methods to the problem of target surveillance under ballistic threat.

The problem of optimally placing sensors in static networks has been extensively studied (Dhillon and Chakrabarty 2003; Akbarzadeh et al. 2013). Recent work extended the problem to environments with static ballistic threats (Richie et al. 2013). These works focus on stationary sensors that do not require a control policy to operate. In the case of adversarial target surveillance, the surveillance agent is dynamic and must move in order to be effective. Dynamic surveillance has also been a topic of previous research. Control policies have been developed for sensor network coverage with guaranteed collision avoidance (Hussein and Stipanovic 2007), target tracking (Lee et al. 2003), and persistence surveillance (Nigam et al. 2012). These methods develop control policies from first principles and provide some optimality guarantees on the solution. However, these approaches do not consider the uncertainties that arise from operating in urban environments filled with line-of-sight obstructions. POMDPs have been applied to problems in which an agent must track a moving target in an environment with line-of-sight obstructions (Hsu, Lee, and Rong 2008), and this paper aims to build on that work.

There are many ways to model the behavior of an adversarial entity in a decision process. One approach is to formulate the problem as a Bayesian Stackelberg game (Paruchuri 2008), in which the agent is uncertain about the types of

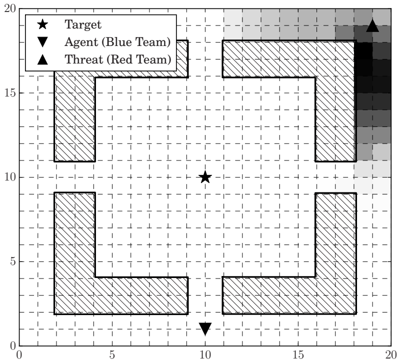

Figure 1: A heat map showing the belief over threat location

adversaries it may face. One of the limitations of Stackleberg games is that they do not provide a structured way of incorporating state uncertainty into the solution. To account for state uncertainty, the problem can be formulated as a multi-agent POMDP with agents that have competing objectives. Adversarial behavior can be incorporated into the POMDP framework using a game-theoretic approach with partially observable stochastic games (Bernstein et al. 2004), or by considering a belief over the models of the other agents with interactive POMDPs (I-POMDPs) (Gmytrasiewicz and Doshi 2005). However, these methods prove intractable for problem sizes considered in this work. An alternative approach to planning in environments with self-interested agents using level-k MDP policies has been shown to improve computational efficiency (Hoang and Low 2013). This approach is based on a quantal level-k model for strategic reasoning, which has been shown to predict human behavior better than other behavior models (Wright and Leyton-Brown 2010). We adopt a similar approach to model the behavior of a stochastic, human adversary.

In this work, we introduce a target surveillance problem with an adversarial ballistic threat whose actions are both intelligent and stochastic. We extend the state-of-the-art approach for solving POMDPs with adversarial agents to a large scale, real world system. Finally, we evaluate our approach with simulations against an adversary that follows a POMDP policy and with a computer game against human players. The approach is shown to effectively model adversarial behavior in POMDPs and to outperform human players in computer game evaluations.

2 Background

This section outlines POMDPs, their solution methods, and nested policies.

2.1 POMDPs

An MDP is a stochastic process in which an agent chooses an action according to its current state in order to maximize the cumulative reward it receives over time. POMDPs are MDPs in which the state of the system is not known, and the agent must rely observations to gather information about the current state of the system. Discrete POMDPs can be represented as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{Z}, T, R, O)$, where:

- $\mathcal{S}$ is the set of partially observable states $s \in \mathcal{S}$;
- $\mathcal{A}$ is the set of possible actions $a \in \mathcal{A}$;
- $\mathcal{Z}$ is the set of observations $o \in \mathcal{Z}$;
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the stochastic transition model;
- $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward model;
- $O : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to [0, 1]$ is the observation model;

At each time step, the agent updates its belief $b(s)$, which defines the probability of being in state s according to its history of actions and observations. Systems with discrete beliefs can be updated exactly using

$$b'(s') \propto O(s', a, o) \sum_{s \in \mathcal{S}} T(s, a, s') b(s). \tag{1}$$

where b' is the new belief after taking a and observing o.

The solution to a POMDP is a policy, or a strategy, that selects actions based on the uncertainty of the underlying system state as well as the uncertainty of the system dynamics. The policy can be represented as a collection of alpha-vectors denoted Γ. Associated with each of the alpha-vectors is one of the actions. The optimal value function for a POMDP can be approximated by a piecewise-linear convex function that takes the form

$$V(b) = \max_{\alpha \in \Gamma} (\alpha \cdot b). \tag{2}$$

If an action associated with an alpha vector α maximizes the inner product $\alpha \cdot b$, then that action is optimal.

2.2 MOMDPs

The POMDP formulation can be extended to problems with mixed observability by using a factored model to separate the fully and partially observable components of the agent's state. This reformulation is known as a mixed observability MDP (MOMDP). It has been shown that, in problems with mixed observability, MOMDPs can be solved more efficiently than POMDPs (Ong et al. 2009). In a MOMDP, the joint state space is factored $\mathcal{S} = \mathcal{X} \times \mathcal{Y}$. The $x \in \mathcal{X}$ variable represents the fully observable state components while the $y \in \mathcal{Y}$ variable represents the partially observable ones. The MOMDP model can be described by the tuple $(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{Z}, T_x, T_y, R, O)$, which differs from the POMDP tuple in its factored representations of the state space and the transition function. Since the state variable x is fully observable, a belief only needs to be maintained over the partially observable variable y. The MOMDP model leads to improvements in computational complexity due to the reduced dimensionality of the belief space.

2.3 POMDP Solution Methods

In general, computing an optimal policy for POMDPs is intractable (Papadimitriou and Tsitsiklis 1987). However, a number of approximate solution methods exist that can scale to large problems (Kochenderfer 2015). A simple approximation method known as QMDP uses the state-action value function $Q(s, a)$ to approximate the alpha vectors. While QMDP performs well in many real-world problems, the method tends to have difficulty with information gathering actions because it assumes perfect state observability after performing the action (Hauskrecht 2000). Point-based methods like Point-Based Value Iteration (Pineau, Gordon, and Thrun 2003) and Heuristic Search Value Iteration (Smith and Simmons 2005) have received attention because of their ability to solve relatively large problems. The Successive Approximation of the Reachable Space under Optimal Policies (SARSOP) algorithm (Kurniawati, Hsu, and Lee 2008) is the current state-of-the-art point-based POMDP solver. The advantages of SARSOP are as follows: i) it explores only the optimally reachable part of the belief space to improve computational efficiency, ii) it provides bounds on the quality of the solution, and iii) it can handle problems with mixed observability. QMDP can be used to solve MOMDPs by assuming all the state variables are partially observable, while SARSOP can handle the MOMDP factorization directly (Ong et al. 2009).

2.4 Level-k Policies

In competitive games, each player's decision making strategy is influenced by the actions of other players. One way to formulate player decision making is though level-k reasoning, which is inspired by the cognitive hierarchy model of games (Camerer, Ho, and Chong 2004). This paper uses a two-agent framework that involves a recursive reasoning process with k levels. At level 0, the agent follows a random policy. At higher levels of reasoning $k \geq 1$, the agent picks a strategy assuming the other agent follows a policy based on a lower level of reasoning $k-1$.

A nested MDP is a multi-agent planning model that considers the intentions of others agents when determining our agent's optimal policy (Hoang and Low 2013). For simplicity, we consider a two agent framework. For a given agent at level k reasoning, a nested MDP assumes that the strategy of the other agent is based on lower levels $0, 1, ..., k-1$ of reasoning. The framework requires a decomposition of the state space to independently describe the dynamics of each agent, a set of k nested policies for the other agent and reward models for both agents. By treating the intentions of the other agent as a stochastic process in which it samples actions from the set of nested policies $\{\pi^0, \pi^1, ..., \pi^{k-1}\}$ with probability p_i we can capture its behavior in the transition model of the nested MDP. The nested MDP can be solved recursively by starting at level 0 reasoning and incrementally solving for the policy of each agent until the desired level of reasoning is reached. This formulation assumes a fully observable state space, allowing each nested policy to be efficiently solved using value iteration (Kochenderfer 2015). In this work, the adversary was modeled using nested level-k policies.

3 Problem Formulation

This section describes the problem of target surveillance under ballistic threat. The problem is modeled as a MOMDP with a level-k nested adversarial policy represented as process noise in the partially observable variable. Model parameters are discussed in this section as well as a heuristic policy used as a baseline in the empirical evaluation.

3.1 Agents and State Spaces

The agents in the model are the Red Team (the ballistic threat) and the Blue Team (the surveillance resource). The target is assumed to be stationary and is not considered a part of the system state. This framework can be extended to a three agent scenario with a dynamic target, but the approach is not explored here. The objective of the Blue Team is to monitor the target while avoiding the Red Team. The objective of the Red Team is to prevent the Blue Team from observing the target. We define a line-of-sight encounter between the two teams as a ballistic engagement. During a ballistic engagement, the Red Team can land a ballistic hit on the Blue Team according to a probability that follows the ballistics model for long range assault weapons (Richie et al. 2013). The ballistic model is a decaying, fifth-order polynomial that depends only on the Euclidean distance between the two teams. In this work, all simulations terminate if the Red Team lands a ballistic hit on the Blue Team. In the MOMDP framework, the position of the Blue Team is represented by the fully observable variable $x_{\text{blue}} \in \mathcal{X}_{\text{blue}}$, while the position of the Red Team is represented by the partially observable variable $x_{\text{red}} \in \mathcal{X}_{\text{red}}$. The complete system state is given by $s = (x_{\text{blue}}, x_{\text{red}})$. The state space is four-dimensional (two dimensions for each position) and is modeled as a grid.

3.2 Action and Observation Spaces

The MOMDP actions control the Blue Team, while the dynamics of the Red Team are modeled as process noise that follows the nested MDP policy described in Section 2.4. Each team can remain stationary, or move to one of the eight neighboring spaces.

The observation space is the union of the Red Team states and a NULL observation. The observation is NULL when the two teams do not see each other or it corresponds to the position of the Red Team when the two teams are within line-of-sight. The size of the observation space is $|\mathcal{Z}| = |\mathcal{X}_{\text{red}}| + 1$.

3.3 Reward Model

The reward function formalizes the objectives of the Blue and Red Teams. The Blue Team is penalized for moving, rewarded for observing the target, and penalized for being within line-of-sight of the Red Team. On the other hand, the Red Team is penalized for moving, penalized when the Blue Team observes the target, and rewarded for a ballistic engagement with the Blue Team. The reward scalars used in this work are summarized in Table 1. Observation-dependent rewards were determined using ray-tracing line-of-sight calculations. The line-of-sight rewards were further weighted according to the ballistics model developed for long range assault weapons (Richie et al. 2013).

3.4 Transition and Observation Models

The state of the Blue Team is the fully observable position of the surveillance agent and transitions according to $T_{\text{blue}}(x_{\text{blue}}, a, x'_{\text{blue}}) = P(x'_{\text{blue}} \mid x_{\text{blue}}, a)$. The transition model for the Red Team, on the other hand, depends on the level of reasoning being considered. The state of the Red Team is the partially observable position of the adversary

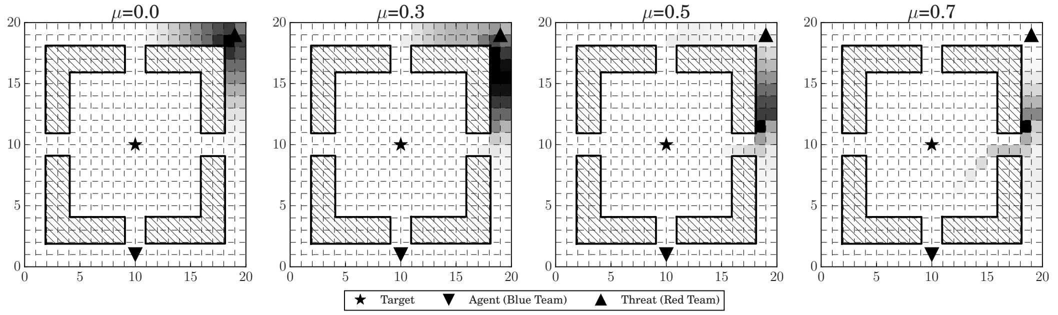

Figure 2: An example instance showing evolution of belief over 15 time steps with varying μ

which transitions according to a policy that is either random ($k = 0$) or pre-computed ($k \geq 1$). The adversary's transition model is described in more detail in Section 3.5.

The observation model relies on ray-tracing calculations to determine if the two teams are within line-of-sight. The observation is NULL when the two teams are not within line-of-sight. The model takes the form $O(x'_{\text{blue}}, x'_{\text{red}}, o, a) = P(o \mid x'_{\text{blue}}, x'_{\text{red}}, a)$.

3.5 Adversary Model and Policy Computation

We extend the formalism of nested policies described in Section 2.4 to model an intelligent adversary in partially observable settings. We define the transition model of the adversary as a set of nested level-$(k-1)$ policies that were computed recursively. To simplify the model, we assume that the adversary follows either the level-$(k-1)$ policy with probability μ or the level-0 policy with probability $(\mu - 1)$. Here μ is defined to be the adversarial stochasticity constant. Specifically, the adversary either follows its π^{k-1} policy or moves randomly to a neighboring space. In this work, the stochasticity constant was chosen to be $\mu = 0.4$, because the resulting adversarial strategy showed expected behavior of an enemy in simulation. Ideally, μ would be chosen with the help of subject matter experts or learned from real adversary actions in surveillance situations. The levels of the Blue and Red Teams were chosen to be three and two respectively. Model parameters for both the adversary and the agent are given in Table 1.

For a given agent, the problem formulation requires the transition model of the opponent. This framework allows us to construct and evaluate both the surveillance and the threat policies by using nested level-k reasoning to model the dynamics of the opponent. Once the transition model of the opponent is constructed, the MOMDP policy can be computed using either QMDP or SARSOP.

An example of an adversarial model is shown in Figure 2. The hashed shapes represent occlusions in the environment, while the dashed lines represent the bins formed by the discretization of the state space. The figure shows the belief of the surveillance agent over the position of the threat after 15 time steps. The surveillance agent knows the location of the threat at $t = 0$, but does not observe it after. The surveillance agent and the threat are both stationary while the belief evolves. The figure demonstrates that a large confidence in the adversarial model can lead to false assumptions if the adversary follows a different behavioral model than the one assumed. However, placing some weight on the value of the policy, leads to overall better performance. In this example, the surveillance agent has a high confidence that the threat has moved down the hallway when $\mu = 0.7$, despite the fact that the threat stayed stationary.

Table 1: Model reward scalars and stochasticity constants

Model	Moving	Observe Target	Ballistic Engagement	μ	k
Threat	-0.01	-0.1	1.0	0.4	2
Agent	-0.01	0.1	-1.0	0.4	3

3.6 Heuristic Policy

A heuristic surveillance policy was developed to serve as a baseline to the POMDP solutions. When the system state is known, the heuristic policy moves the Blue Team towards the position with the largest immediate reward, and assumes that the Red Team is stationary. In case of ties, the closest position is chosen. The position with the largest reward is determined using local search. In the case of partial observability, the location of the Red Team is determined by taking the maximum over the belief vector. The shortcomings of the heuristic is that it follows a greedy policy that does not take into account future state information. It also relies on a naive approach to choosing the approximate system state.

4 Results

This section provides empirical evaluations of the SARSOP and QMDP surveillance policies and compares them against a random policy, the heuristic policy, and the fully observable policy. The random policy uniformly samples allowable actions, while the fully observable policy assumes perfect knowledge in the position of the other team and was obtained using value iteration (Kochenderfer 2015). The evaluation was performed on three different urban environments discretized on 20 × 20 grids: Map A, Map B and Map C shown in Figure 3. The primary goals of the empirical analysis are: i) to evaluate the performance of each policy, ii) to analyze the level-k model of the ballistic threat, and iii) to evaluate the performance of the policies against a human.

The initial conditions for each simulation were set using rejection sampling. The state space was sampled uniformly, and states were rejected if the Blue and Red Teams were within line-of-sight or if the Blue Team was within line-of-sight of the target. Rejection sampling ensured a realistic starting scenario for the simulations. Initial beliefs were set to be uniform for policies with partial observability. Each simulation ran for 100 time-steps or until the Red Team landed a ballistic hit on the Blue Team.

4.1 Solution Comparison

Table 3 shows the average rewards for five surveillance policies on Map B over 500 simulations. The policies are compared against an adversary that follows a level 2 SARSOP policy, a heuristic policy, and a random policy. The results show that the SARSOP surveillance policy outperforms all other policies in a partially observable environment. The QMDP policy suffers from inability to perform information gathering actions, and at times leads to a stationary agent that fails to evade the threat and to observe the target. The heuristic policy assumes the system state to be the one with highest belief probability, and often follows a policy based on the incorrect position of the adversary, leading to poor performance. The nested MDP approach has the best performance, but makes the unrealistic assumption that the position of the adversary is fully observable at all times. The results from the random and the nested MDP policies set a baseline and an upper bound, respectively, on the performance of the surveillance agent. All surveillance policies perform significantly worse against the SARSOP adversary than against either the heuristic or the random one. The compute times for each policy are shown in Table 2.

Figure 3 shows heat maps of the surveillance agent's location after a ballistic hit by the threat for the SARSOP and the heuristic surveillance policies. One difference between the two policies is that ballistic engagements for SARSOP primarily occur in a few locations on the grid, while the engagements for the heuristic are more uniformly spread out throughout the map. This qualitative difference is indicative of the overall performance of each policy, as an intelligent agent is more likely to stay in one location from which a large portion of the map can be seen while still observing the target. On Map A, for example, the two locations with the largest number of ballistic engagements have line-of-sight coverage of large portions of the map (including the target), and allow for a quick escape from the threat if it appears from either side of the covered portion of the map.

Average survival probabilities are shown in Figure 4. The survival probability indicates the probability of the surveillance agent avoiding a ballistic hit throughout the 100 time-step simulation. The survival probabilities against a random adversary are significantly higher than those against an intelligent adversary that follows a SARSOP policy. Survival probabilities on maps A and C are lower against an intelligent adversary because of the open structure of those maps.

Table 2: Policy compute times

	Heuristic	QMDP	SARSOP
Compute Time	0.1 sec	60 sec	15 min

Table 3: Average rewards for five level 3 surveillance policies

	Threat Policy		
	SARSOP	Heuristic	Random
Random	-0.69 ± 0.2	-0.38 ± 0.2	-0.06 ± 0.1
Heuristic	-0.21 ± 0.1	0.55 ± 0.2	0.86 ± 0.4
QMDP	0.32 ± 0.4	1.08 ± 0.5	1.52 ± 0.2
SARSOP	0.96 ± 0.4	2.88 ± 0.5	4.23 ± 0.4
Nested MDP	2.30 ± 0.4	3.65 ± 0.2	5.01 ± 0.2

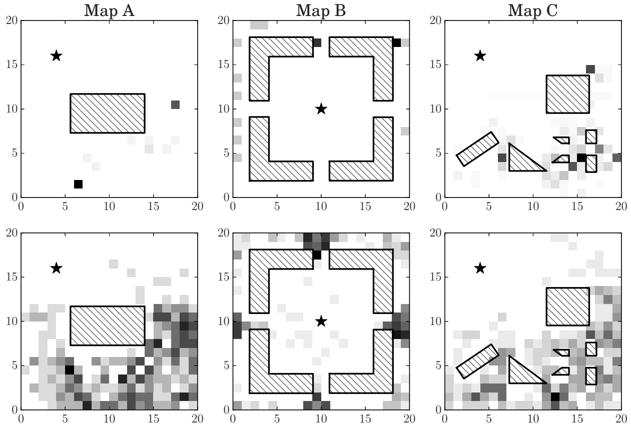

Figure 3: Heat maps showing the last location of the surveillance agent after a ballistic hit over 500 simulations for the level 3 SARSOP policy (top) and the heuristic policy (bottom) against a level 2 SARSOP adversary on Map A (left), Map B (middle), and Map C (right). The star indicates the location of the target.

Figure 5 shows the performance of level-k surveillance policies evaluated against level-k policies of the threat. The curves track the average reward for a given reasoning level of a threat. The performance of the agent against a threat following a random (level 0) policy stays constant across different levels of reasoning. However, there are subtle improvements in performance of level-k agents against level-$(k-1)$ adversaries. On average, level 3 surveillance policies outperform all other levels of reasoning when the level of the adversary may be unknown.

4.2 Games Against Humans

A video game was developed for the purpose of evaluating the surveillance policies against a human opponent. A screen shot of the game is shown in Figure 6. Nearest neighbor interpolation was used to select discrete actions in the continuous video game domain. The human player controlled the surveillance agent against an adversary following a level 2 SARSOP policy. The players were given the objectives of the game as well as an unlimited number of trial runs. After the players

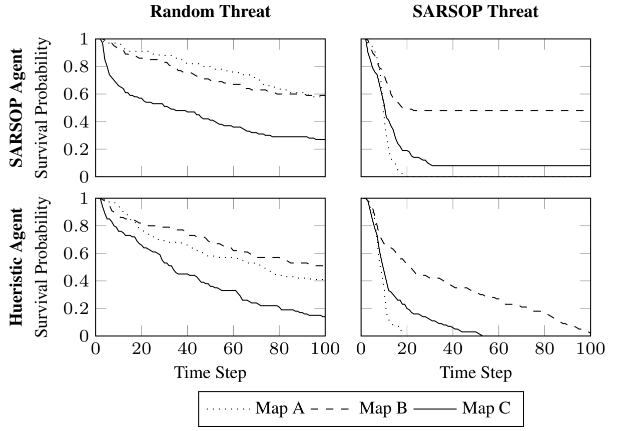

Figure 4: Survival probabilities for a surveillance agent following level 3 SARSOP and heuristic policies against a random and a level 2 SARSOP threat. Each curve was generated using 500 simulations.

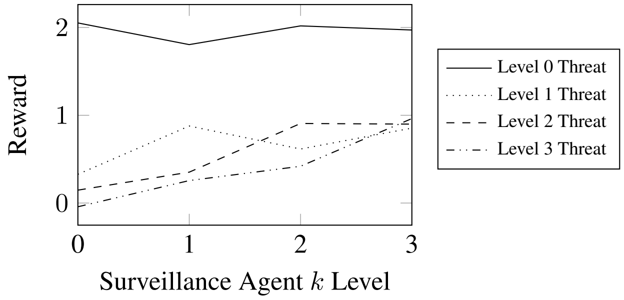

Figure 5: Average rewards for level-k SARSOP surveillance policies against level-k SARSOP threat policies

felt comfortable with the game controls, the results of their first ten play outs were recorded. Figure 7 shows a performance curve for level 3 and level 1 SARSOP policies against a level 2 SARSOP adversary generated by varying the reward of observing the target from zero to one while keeping other reward parameters fixed. The average results for human players over the ten play outs are shown in the figure as well. The performance curves show that human players have a difficult time exceeding or meeting the level of survivability achieved by the SARSOP surveillance policies while observing the target at similar rates. The level 3 SARSOP performance curve dominates the level 1 curve. The curves demonstrate that the level-k policies lead to improved performance over naive policies that assume a randomly moving adversary.

5 Conclusion

Despite the advances in optimal control and automated planning over the recent years, the design of a robust target surveillance system that accounts for ballistic adversaries remains a challenge. The difficulties lie in accounting for an unpredictable adversary and the line-of-sight occlusion that lead to partial observability of the system state. We modeled adversarial target surveillance as a MOMDP, where the adversary's actions followed a level-k decision-making policy, and solved the problem using QMDP and SARSOP. We showed this approach to work well in simulation and to outperform a human player in a video game.

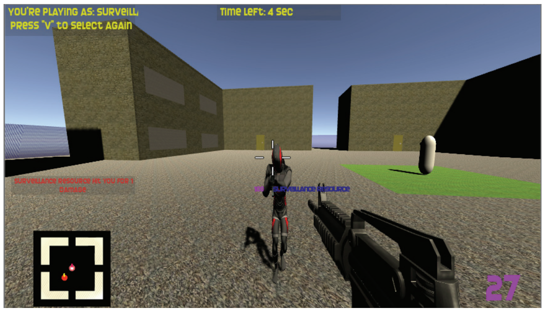
YOU'RE PLAYING AS: SURVEILL
PRESS "V" TO SELECT AGAIN
TIME LEFT: 4 SEC
SURVEILLANCE RESOURCE HIT YOU FOR 1 DAMAGE
100
SURVEILLANCE RESOURCE
27

Figure 6: A screen shot of the video game used to evaluate target surveillance policies

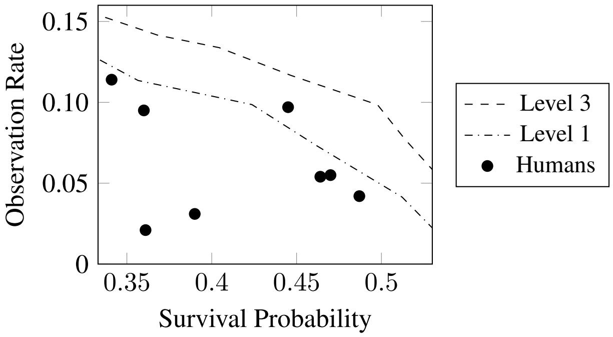

Figure 7: Performance curves for the target surveillance system following SARSOP level 3 and level 0 policies against a level 2 SARSOP adversary. Average performance of eight human players over ten games are plotted as well.

There are several areas for future work. Extending POMDP based surveillance planners to real world applications requires dealing with continuous domains. One approach is to use Monte Carlo Value Iteration (MCVI) (Bai et al. 2010) to solve continuous state POMDPs. MCVI has been successfully applied to complex problems such as unmanned aircraft collision avoidance (Bai et al. 2011), and may be applied to the continuous state target surveillance framework as well. Another area of future work involves generalizing surveillance plans to new environments. One way to approach this problem is through deep reinforcement learning (DRL), where a neural network serves as a value function approximator and Q-learning is used to train the network. Recently, DRL has been successfully applied to playing Atari games (Mnih et al. 2015). A similar approach can be taken in the surveillance problem, where in addition to the state, reward and action information, the input to the neural network consists of the structure of the surveillance environment.

The source code for this work can be found at https://github.com/sisl/TargetSurveillance.

6 Acknowledgments

This work was supported by the Army AHPCRC grant W911NF-07-2-0027. The authors would like to thank the anonymous reviewers for their helpful comments.

References

Akbarzadeh, V.; Gagne, C.; Parizeau, M.; Argany, M.; and Mostafavi, M. 2013. Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage. *IEEE Transactions on Instrumentation and Measurement* 62:293–303.

Bai, H.; Hsu, D.; Lee, W. S.; and Ngo, V. A. 2010. Monte Carlo value iteration for continuous-state POMDPs. In *Workshop on the Algorithmic Foundations of Robotics*.

Bai, H.; Hsu, D.; Kochenderfer, M. J.; and Lee, W. S. 2011. Unmanned aircraft collision avoidance using continuous-state POMDPs. In *Robotics: Science and Systems*.

Bernstein, D. S.; Hansen, E. A.; Bernstein, D. S.; and Amato, C. 2004. Dynamic programming for partially observable stochastic games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 709–715.

Camerer, C. F.; Ho, T.-H.; and Chong, J.-K. 2004. A cognitive hierarchy model of games. *Quarterly J. Economics* 119:861–134.

Dhillon, S. S., and Chakrabarty, K. 2003. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *IEEE Transactions on Wireless Communication and Networking*, 1609–1614.

Gmytrasiewicz, P. J., and Doshi, P. 2005. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research* 24:49–79.

Hauskrecht, M. 2000. Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research* 13:33–94.

Hoang, T. N., and Low, K. H. 2013. Interactive POMDP Lite: Towards practical planning to predict and exploit intentions for interacting with self-interested agents. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2298–2305.

Hsu, D.; Lee, W. S.; and Rong, N. 2008. A point-based POMDP planner for target tracking. In *IEEE Conference on Robotics and Automation*, 2644–2650.

Hussein, I. I., and Stipanovic, D. M. 2007. Effective coverage control for mobile sensor networks with guaranteed collision avoidance. *IEEE Transactions on Control Systems Technology* 15:642–657.

Kochenderfer, M. 2015. *Decision Making Under Uncertainty: Theory and Application*. MIT Press.

Kurniawati, H.; Hsu, D.; and Lee, W. S. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*.

Lee, J.; Huang, R.; Vaughn, A.; Xiao, X.; Hedrick, J. K.; Zennaro, M.; and Sengupta, R. 2003. Strategies of path-planning for a UAV to track a ground vehicle. In *Autonomous Intelligent Networks and Systems Conference*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Nigam, N.; Bieniawski, S. R.; Kroo, I.; and Vian, J. 2012. Control of multiple UAVs for persistent surveillance: Algorithm and flight test results. *IEEE Transactions on Control Systems Technology* 20(5):1236–1251.

Ong, S. C. W.; Png, S. W.; Hsu, D.; and Lee, W. S. 2009. POMDPs for robotic tasks with mixed observability. In *Robotics: Science and Systems*.

Papadimitriou, C., and Tsitsiklis, J. N. 1987. The complexity of Markov decision processes. *Mathematics of Operations Research* 12(3):441–450.

Paruchuri, P. 2008. Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Pineau, J.; Gordon, G.; and Thrun, S. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1025 – 1032.

Richie, D. A.; Ross, J. A.; Park, S. J.; and Shires, D. R. 2013. Ray-tracing-based geospatial optimization for heterogeneous architectures enhancing situational awareness. *2013 IEEE 16th International Conference on Computational Science and Engineering* 81–86.

Smith, T., and Simmons, R. G. 2005. Point-based POMDP algorithms: Improved analysis and implementation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 542–547.

U.S. Army. 2008. *Campaign Planning Handbook*.

Wright, J. R., and Leyton-Brown, K. 2010. Beyond equilibrium: Predicting human behavior in normal-form games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 901–907.