Motion Strategies for Surveillance

Sourabh Bhattacharya Salvatore Candido Seth Hutchinson
Department of Electrical and Computer Engineering
University of Illinois at Urbana Champaign
Urbana, Illinois
Email: {sbhattac, candido, seth}@uiuc.edu

***Abstract*— We address the problem of surveillance in an environment with obstacles. We show that the problem of tracking an evader with one pursuer around one corner is completely decidable. The pursuer and the evader have complete information about each other's instantaneous position. The pursuer has complete information about the instantaneous velocity of the evader. We present a partition of the visibility region of the pursuer where based on the region in which the evader lies, we provide strategies for the evader to escape the visibility region of the pursuer or for the pursuer to track the target for all future time. We also present the solution to the inverse problem: given the position of the evader, the positions of the pursuer for which the evader can escape the visibility region of the target. These results have been provided for varying speeds of the pursuer and the evader. Based on the results of the inverse problem we provide an $O(n^3 \log n)$ algorithm that can decide if the evader can escape from the visibility region of a pursuer for some initial pursuer and evader positions. Finally, we extend the result of the target tracking problem around a corner in two dimensions to an edge in three dimensions.**

I. Introduction

Surveillance is related to target tracking and the game of pursuit-evasion. The goal of the pursuer is to maintain a line of sight to the evader that is not occluded by any obstacle. The goal of the evader is to escape the visibility polygon of the pursuer (and break this line of sight) at any instant of time.

This problem has several interesting applications. It may be useful for a security robot to track a malicious evader that is trying to escape. The robot must maintain visibility to ensure the evader will not slip away while another party or the pursuer itself attempts to eventually trap or intercept the evader. Also, an "evader" may not be intentionally trying to slip out of view. A pursuer robot may simply be asked to continuously follow and monitor at a distance an evader performing a task not necessarily related to the target tracking game. The pursuer may somehow be supporting the evader or relaying signals to and from the evader. The pursuer may also be monitoring the evader for quality control, verifying the evader does not perform some undesired behavior, or ensuring that the evader is not in distress. Finally, the results are useful as an analysis of when escape is possible. If it is impossible to slip away, it may be desirable for the evader to immediately surrender or undertake a strategy not involving escape.

A great deal of research exists on pursuit-evasion. Pursuit-evasion games are analyzed in $\mathbb{R}^n$ [19], in non-convex domains of arbitrary dimension [1], and in graphs [34], [33]. A large volume of game theoretic formulations and analysis can be found in [2], [15], [35] and [21]. Also, [9] presents an approach that takes into account the pursuer's positioning uncertainty. While this analysis is pertinent, it often focuses on "capturing", moving within a certain distance, of the evader. We only seek to maintain a line of sight to the evader.

A related but different problem in the robotics community is to find an evader with one or more pursuers in various environments. Exact [12], [23], [24], [25], [38], [40], [11] and probabilistic [13], [41] strategies have been found to locate an evader while preventing it from slipping into a region that has already been searched. Randomized strategies have been employed to locate and capture an unpredictable evader in any simply connected polygon [17] and [16]. [39] deals with one pursuer searching one evader with k flashlights in a general environment. The paper presents necessary and sufficient conditions for various searchers. Our problem assumes the evader starts in a position visible to the pursuer. The pursuer's goal is to track the evader rather than search the environment for a hidden evader.

The problem of maintaining visibility of a moving evader has been traditionally addressed with a combination of vision and control techniques [8], [26], [27], and [14]. Pure control approaches are local by nature and do not take into account the global structure of the environment. Our interest is in deriving pursuer strategies that guarantee successful surveillance taking into account both constraints on motion due to obstacles and constraints on visibility due to occlusion.

The problem of target tracking has also been analyzed at a fixed distance between the pursuer and evader. In [28] and [29], optimal motion strategies are proposed for a pursuer and evader based on critical events and in [30] a target tracking problem is analyzed with delay in sensing. These papers are summarized in [32]. [3] deals with the problem of *stealth target tracking* where a robot equipped with visual sensors tries to track a moving target among obstacles and, at the same time, remain hidden from the target. Obstacles impede both the tracker's motion and visibility, and also provide hiding places for the tracker. A tracking algorithm is proposed that applies a local greedy strategy and uses only local information from the tracker's visual sensors and assumes no prior knowledge of target tracking motion or a global map of the environment.

[20] presents a method of tracking several evaders with multiple pursuers in an uncluttered environment. In [18] the problem of tracking multiple targets is addressed using a network of communicating robots and stationary sensors. A

region-based approach is introduced which controls robot deployment at two levels, namely, a coarse deployment controller and a target-following controller.

In [22] the problem of computing robot motion strategies that maintain visibility of a moving target is studied under deterministic and stochastic settings. For the deterministic target, an algorithm that computes optimal, numerical solutions is presented. For the stochastic case, two online algorithms are presented that each attempt to maintain future visibility with limited perception. In [31] one or more observers try to track one or more targets by maximizing the shortest distance the target needs to move in order to escape the observer's visibility region. In [10][4] a target tracking problem is analyzed for an unpredictable target and an observer lacking prior model of the environment. It computes a risk factor based on the current target position and generates a feedback control law to minimize it. [7] presents an algorithm that maximizes the evader's minimum time to escape for an evader moving along a known path.

There have been successful efforts in the past to deploy tracking and surveillance systems in real world. We only present a few of them. [37] and [36] developed a distributed heterogeneous robotic team that is based mainly on a miniature robotic system. Most of the robots are extremely small because some operations require covert action. They team the scouts with larger ranger robots that can transport the scouts over distances of several kilometers, deploy them rapidly over a large area, coordinate their behavior, and collect and present the resulting data. [5] presents a mobile robot called the *Intelligent Observer* which moves through an environment while autonomously observing moving targets selected by a human operator. The robot carries one or more cameras which allow it track objects while at the same time sensing its own location.

We address the problem of target tracking in an environment with one corner, one pursuer, and one evader. This is the first result, to our knowledge, that the surveillance problem is decidable around one corner and gives partitions of the workspace that demonstrate the outcome of the game. While the general problem of deciding whether the evader can escape or the pursuer can track the evader forever in any arbitrary polygonal environment is still, so far as we know, an open problem, we offer partial solutions to two important problems. First, we provide sufficient conditions for escape. These conditions could be used to solve the evader's problem (i.e., he would construct an escape strategy that exploited these conditions) when they are satisfied, thus providing a partial solution to this pursuit-evasion problem. Second, our analysis is in the direction of providing open loop policies for the pursuer to track the evader. Closed loop policies for the pursuer depend on the current state of the evader resulting in a delay in the reaction of the pursuer due to the time required to process sensor data. The model changes from a continuous time system to a discrete time system that in general lead to computationally intractable algorithms. Moreover due to the delay introduced in processing, the sufficient conditions weaken.

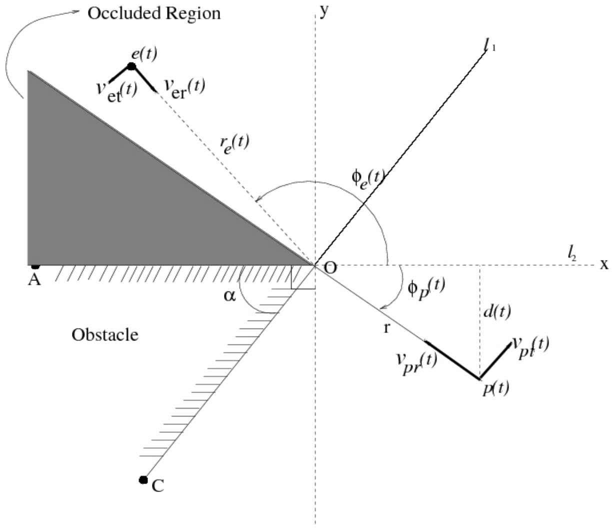


Fig. 1. The problem environment

In Section 2, based on the geometry of the corner, the ratio of maximum pursuer and evader velocities, and the initial position of the pursuer, we segment the free space into regions. The pursuer policy and the possibility of evader escape are determined by the region in which the initial evader position is contained. Pursuer policies are given that guarantee, for some regions of initial conditions, that no evader policy will lead to escape of the evader at any future time. It is then proved that outside these regions, there is an evader policy by which escape is inevitable irregardless of the pursuer policy. In Section 3, the same analysis is performed with respect to the initial position of the evader. In Section 4, we use the results of the previous section to construct a algorithm that can decide whether the evader can escape the pursuer for certain initial positions of the pursuer and the evader. In Section 5, we extend the above results to target tracking in $\mathbb{R}^3$ around an edge.

II. Pursuer-Based Partition

A mobile pursuer and evader exist on a plane at points $p(t)$ and $e(t)$, respectively. They are point robots and move with bounded speeds, $v_p(t)$ and $v_e(t)$. Therefore, $v_p(t) : [0, \infty) \rightarrow [0, \overline{v}_p]$ and $v_e(t) : [0, \infty) \rightarrow [0, \overline{v}_e]$. We assume that $v_p(t)$ and $v_e(t)$ can be discontinuous functions of time.

The workspace contains a semi-infinite obstacle with one corner that restricts pursuer and evader motions and may occlude the pursuer's line of sight to the evader. Without loss of generality, this corner is placed at the origin and one of the sides lies along the -x axis as shown in Figure 1. The unshaded region is the visibility region of the pursuer. $v_{pt}(t)$ and $v_{et}(t)$ describe the pursuer and evader tangential velocities. $v_{pr}(t)$ and $v_{er}(t)$ describe the radial velocities. $\phi_e(t)$ and $\phi_p(t)$ are the angles the evader and pursuer, respectively, make with the +x axis. Note that $\phi_e(t)$ is positive in the counterclockwise direction while $\phi_p(t)$ is positive in the clockwise direction. The minimum distance of the pursuer from line l_2 is denoted

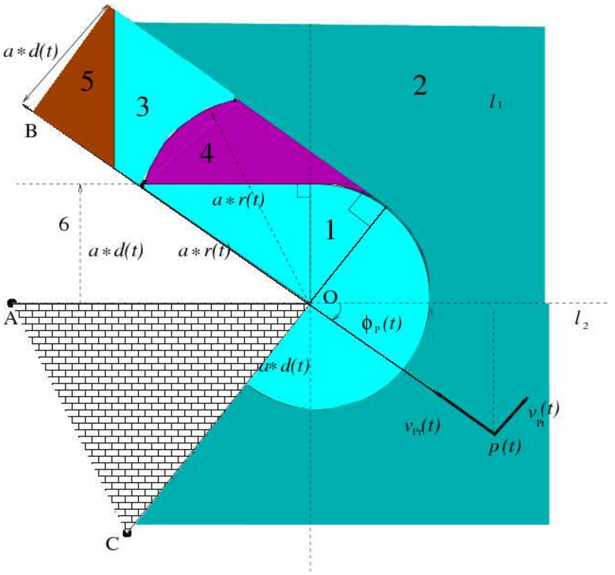


Fig. 2. Pursuer-based partition

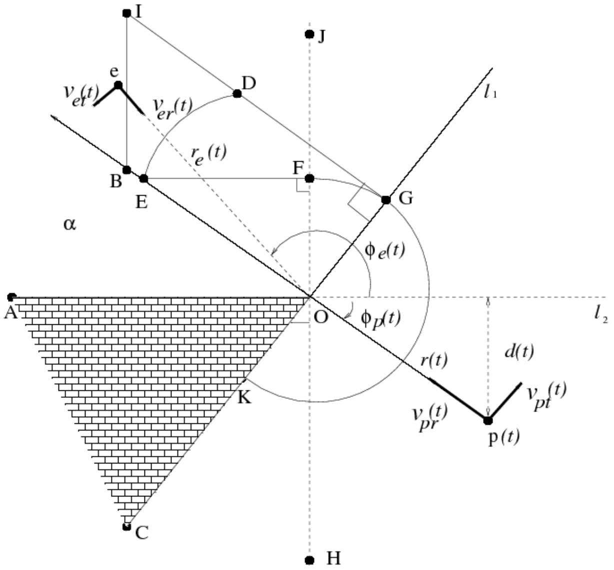


Fig. 3. The geometry of the partition

by $d(t)$. The distance of the pursuer from the corner is denoted by $r(t)$. The distance of the evader from the corner is denoted by $r_e(t)$.

The two edges meeting at this corner are considered to extend for an infinite length so there is no other geometry that the evader can hide behind in the workspace. The two sides of the obstacle form an angle α. If $\alpha \geq \pi$ then every point in the free workspace is visible to every other point and the pursuer will trivially be able to track the evader indefinitely. Thus, we only consider obstacles where $\pi > \alpha \geq 0$.

To prevent the evader from escaping, the pursuer must keep the evader in its visibility polygon, $V(p(t))$. The visibility polygon of the pursuer is the set of points from which a line segment from the pursuer to that point does not intersect the obstacle region. The evader escapes if at *any* instant of time it can break the line of sight to the pursuer. Visibility extends uniformly in all directions and is only terminated by workspace obstacles (omnidirectional, unbounded visibility).

We define the *star region* associated with a corner as the region bounded by the supporting lines of the two edges of a corner. We define the *star region* as the collection of all star points. As can be seen in Figure 1, the star-region extends outward from the corner of the obstacle. It is semi-infinite and bounded by rays l_1 and l_2. From any point in the star-region, the entire free space is visible. If the pursuer can enter the star region before losing sight of the evader, it will trivially be able to track the evader at all future times.

We want to address the following question. Given the initial position of the pursuer and the evader, the map of the environment, $\overline{v}_e$ and $\overline{v}_p$:

1) Does there exist a policy or an algorithm that takes finite steps to provide a policy for the pursuer to track the evader for all future times?
2) Does there exist a policy or an algorithm that takes finite steps to provide a policy for the evader to escape the visibility region in finite time?

We refer to the above questions by the term *decidability*. If the answer to one of the questions is affirmative at every configuration for an environment, we say the problem is *decidable* in that environment. Around one corner the surveillance problem is decidable.

Let $a = \overline{v}_e/\overline{v}_p$, the ratio of maximum evader and pursuer speeds, and define $d = d(t = 0)$. The outcome of the game can be decided based on the velocity ratio and initial positions of the evader and pursuer. This leads to a decomposition of the visibility region of the pursuer into regions in which the evader may lay. Region 1 is the set of all points closer than $a \cdot d(t)$ to segment AO, the far side of the obstacle. Region 2 is the set of points farther away than $a \cdot d(t)$ to segment OB, the edge of $V(p(t))$ laying in free space. Region 3 consists of points laying within distance $a \cdot d(t)$ to segment OB and farther than $a \cdot r(t)$ from point O, the corner. Region 4 is the set of points within distance $a \cdot d(t)$ from segment OB, closer than $a \cdot r(t)$ from point O and farther than $a \cdot d(t)$ from segment AO. Region 5 is the set of points within distance $a \cdot d(t)$ from segment OB and at a distance greater than $\frac{a*r(t)}{\cos(\theta)}$ from the origin. Region 6 is the portion of the free workspace

TABLE I

Evader Policies	Evader Region	Control Law
A	1 and $\phi_e \in [\alpha - \pi, \frac{\pi}{2}]$ 1 and $\phi_e \in [\frac{\pi}{2}, \pi - \phi_p]$	$\dot{r}_e(t) = \overline{v}_e$ $\dot{y}_e(t) = -\overline{v}_e$
Pursuer Policies	Evader Region	Control Law
B	2, 4	$\dot{y}_p(t) = \overline{v}_p$
C	3	$v_{pt}(t) = -\frac{r}{r_e} \lvert v_{et}(t) \rvert$ $v_{pr}(t) = -\frac{r}{r_e} \lvert v_{er}(t) \rvert$
D	5	$v_{pt}(t) = v_p$

not belonging to $V(p(t))$. The pursuer and evader policies necessary to decide the problem can be determined by this partition of $V(p(t))$ shown in Figure 2. These policies are summarized in Table I.

For the remainder of this section, refer to Figures 2 and 3 and Table I. Consider the case where $\alpha < \frac{\pi}{2}$ and $\phi_p \in \left(0, \frac{\pi}{2}\right]$.

Proposition 1 If the evader lies in Region 1 and follows Policy A, no pursuer policy exists that can prevent the escape of the evader.
Proof: If the evader lies in Region 1, the maximum time required by the evader to reach line AO by following Policy A is $t_e < \frac{a \cdot d}{\overline{v}_e} = \frac{d}{\overline{v}_p}$. The minimum time required by the pursuer to reach line l_2 with any policy is at least $t_p > \frac{d}{\overline{v}_p}$. Thus, $t_p > t_e$. Therefore the evader reaches the line AO before the pursuer can reach line l_2. If the evader lies on AO and the pursuer has not yet reached l_2 the evader will be outside the visibility region of the pursuer. Hence the evader escapes. ■

Proposition 2 If the evader lies in Region 2 and the pursuer follows Policy B, no evader policy exists that can escape the visibility region of the pursuer.
Proof: The time required by the pursuer to reach line l_2 by following Policy B is $t_p = \frac{d}{\overline{v}_p}$. If the evader lies in Region 2, the minimum time required by the evader to reach ray OB is $t_e > \frac{a \cdot d}{\overline{v}_e} = \frac{d}{\overline{v}_p}$. Thus, $t_e \geq t_p$. If the pursuer follows Policy B, $V(p(t=0)) \subseteq V(p(t>0))$. Since the evader cannot reach ray OB, the only free boundary of $V(p(t=0))$, before the pursuer reaches the boundary of the star region, $e(t) \in V(p(t)) \forall t \in [0, t_p]$. Once the pursuer reaches the line l_2, the entire free workspace belongs to the $V(p(t_p))$. The pursuer stops hence the evader remains in sight of the pursuer for all future times. ■

Proposition 3 If the evader lies in Region 3 and the pursuer follows Policy C, for every evader policy the evader can either stay in Region 3 or move to region 2 of V$(p(t))$.
Proof: If the pursuer follows Policy C, then it follows both the radial and angular movements of the evader. The geometry of Region 3 is such that $r_e(t) \geq a \cdot r(t)$ so $r(t)/r_e(t) \leq 1/a$. Multiply that with the velocity bound of the evader, $v_e(t) \leq \overline{v}_e$. This quantity is equal to the pursuer velocity of the control law of Policy C.

$$v_p(t) \quad = \quad v_e(t)\frac{r(t)}{r_e(t)} \leq \frac{\overline{v}_e}{a} = \overline{v}_p$$

Thus, the pursuer velocities of Policy C are always attainable in Region 3. In order for the pursuer to maintain sight of the evader, the following equation must hold.

$$\phi_e(t) + \phi_p(t) \quad \leq \quad \pi$$

The tangential component of the control law implies

$$\begin{aligned} \dot{\phi}_e(t) &\leq -\dot{\phi}_p(t) \\ \Rightarrow \dot{\phi}_e(t) + \dot{\phi}_p(t) &\leq 0 \\ \Rightarrow \phi_e(t>0) + \phi_p(t>0) &\leq \phi_e(t=0) + \phi_p(t=0) \leq \pi \ . \end{aligned}$$

Thus, the evader cannot enter Region 5. The radial component of the control law implies

$$\begin{aligned} \frac{\dot{r}_e(t)}{r_e(t)} &= \frac{\dot{r}(t)}{r(t)} \\ \Rightarrow \frac{r_e(t)}{r(t)} &= \frac{r_e(0)}{r(0)} \geq a \end{aligned}$$

Thus, the evader cannot enter Region 4. Hence for any policy the evader can either stay in Region 3 or only enter Region 2. ■

Proposition 4 If the evader lies in Region 4 and the pursuer follows Policy B, for every evader policy the evader can either stay in Region 4 or move to regions 2 or 3 of V$(p(t))$.
Proof: If the pursuer follows Policy B, all points on segment EF move with velocity $a \cdot \overline{v}_p = \overline{v}_e$ toward the ray OA. Similarly, all points on the arc FG move with radial velocity $\overline{v}_e$ toward O. In order to enter Region 1 from Region 4, the evader must move toward the boundary of Region 1 with a velocity greater than the velocity at which the boundary is moving away from the evader. That is not possible since the boundary of Region 1 moves with velocity $\overline{v}_e$, the maximum possible evader velocity, away from the evader. Hence the evader cannot enter Region 1 from Region 4. Hence for all evader policies, the evader can only reach Region 3 or Region 2 from Region 4. ■

Proposition 5 For all initial positions of the evader in Regions 3 and 4, the pursuer can track of the evader by following a reactive motion and switching between policies B and C appropriately.
Proof: If the evader starts in Region 3 and remains in Region 3 then we have proved in Proposition 3 that Policy C for the pursuer can keep the evader in sight for all future time. While the pursuer is following policy C, if the evader enters Region 2, by Proposition 2, the pursuer can track the evader indefinitely by following Policy B. Hence the pursuer can keep sight of the evader for all future time.

If the evader starts in Region 4 and the evader remains in Region 4 then Proposition 4 proves that Policy B for the pursuer can keep the evader in sight for all future time. While the pursuer is following policy B, if the evader moves to Region 3, the strategy provided in the previous paragraph can keep the evader in sight for all future times. While the pursuer is following policy B, if the evader moves to Region 2, by Proposition 2, the pursuer can indefinitely track the evader by following Policy B. Thus, the pursuer will keep the evader in sight for all future time. ■

Proposition 6 For all initial positions of the evader in region 5, the pursuer can track the evader by following policy D.
Proof: Refer to Figure 4. After time t, the evader lies in the closure of a circle of radius $\overline{v}_e t$ centered at $e(0)$. A sufficient condition for the pursuer to keep the evader in sight for all future times is to keep the angular velocity of the line of the sight, OP, to be greater than the angular velocity of the line tangent to the growing circle, OL, for all future time until the

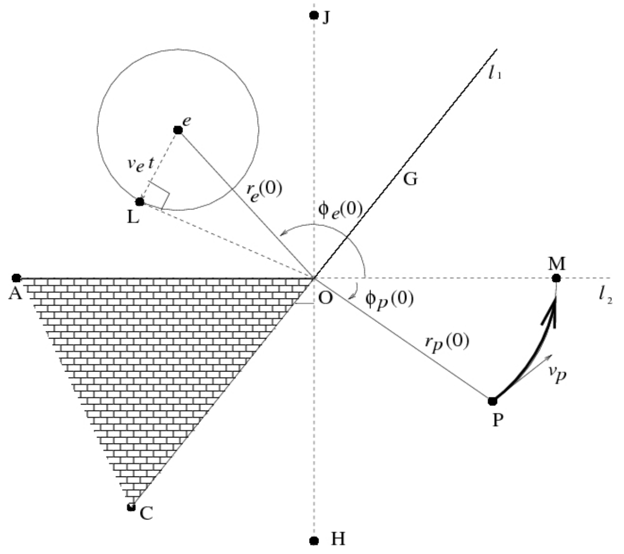


Fig. 4. Evader in region 6

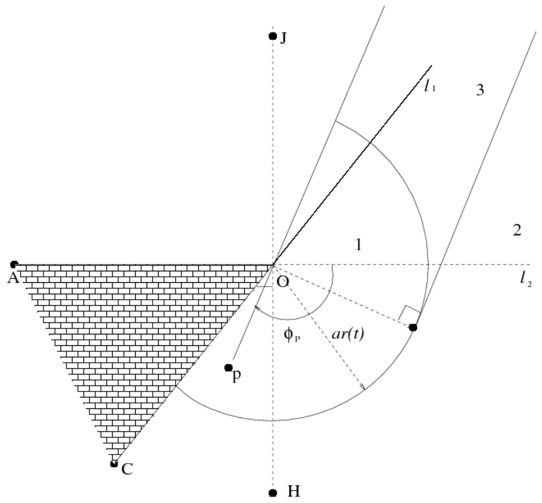


Fig. 5. Pursuer-based partition for the pursuer in region COH

pursuer reaches the *star region*. The angular velocity of the line OP is given by $\omega_p = \frac{v_p}{r_p}$. The maximum angular velocity of the evader is given by $\omega_{emax} = \frac{-v_e}{r_e(0)\cos(\phi_e(0))}$. Solving for $\omega_p \geq \omega_{emax}$ leads to the following condition

$$r_e(0) \geq -\frac{a * r_p}{\cos(\phi_e(0))}$$

which is satisfied for all points in Region 5. ■

If $\phi_p > \frac{\pi}{2}$ the analysis still holds. The only changes are that Region 1 expands, the area of Region 4 is reduced to zero and Region 5 ceases to exist. Figure 5 shows the partition of the visibility region of the pursuer in this case. Note that if $\alpha \in \left[\frac{\pi}{2}, \pi\right]$, then ϕ_p must be less than $\frac{\pi}{2}$ and this case is not a consideration.

Corollary 1 There exists an evader policy that no pursuer policy can track if and only if the evader lies in Region 1.

Proof: The proof can be concluded from the proofs of Propositions 1, 2 and 5. ■

All the above analysis was done for initial positions of the pursuer outside of the star region. If the initial position of the pursuer is in the star region the entire free space is visible to the pursuer. The policy of the pursuer will be to remain stationary and it will trivially be able to track the evader indefinitely.

III. Evader-Based Partition

In the previous section, a partition of the $V(p(t))$ has been given. From only one region a evader policy exists that allows the evader to escape for any pursuer policy. This section presents a partition of the visibility region of the evader, $V(e(t))$. The regions of the partition determine whether there will be an evader policy that guarantees escape. In short, we address the following question - Given the initial position of the evader, from what pursuer positions will the evader be able to escape as a function of the ratio of their speeds?

To find the set of initial pursuer placements from which the evader may escape we must consider two cases depending on whether the closest point to the evader on the obstacle lies on the corner or belongs uniquely to one of the sides.

Refer to Figure 6(a) where the partition for the case where the closest point is on one of the sides of the obstacle is considered. For this situation to occur the evader must lie outside the shaded region of Figure 6(a). Consider, without loss of generality, the case where the evader lies in quadrant II. It can be concluded from Corollary 1 that if the pursuer lies below l_2, the pursuer must be a distance of at least d_e/a from l_2 for the evader to escape. If the pursuer lies in the region between l_1 and OA, it is possible for the evader to escape to the side of the obstacle opposite the evader, OC. Thus, the distance between the pursuer and the closest point on l_1 must be greater than r_e/a for the evader to escape. If the pursuer lies in the region enclosed by l_1 and l_2, the evader cannot escape as the pursuer already lies in the star region. In summary, an evader policy that guarantees escape exists only if the pursuer lies outside of the shaded region of Figure 6(b).

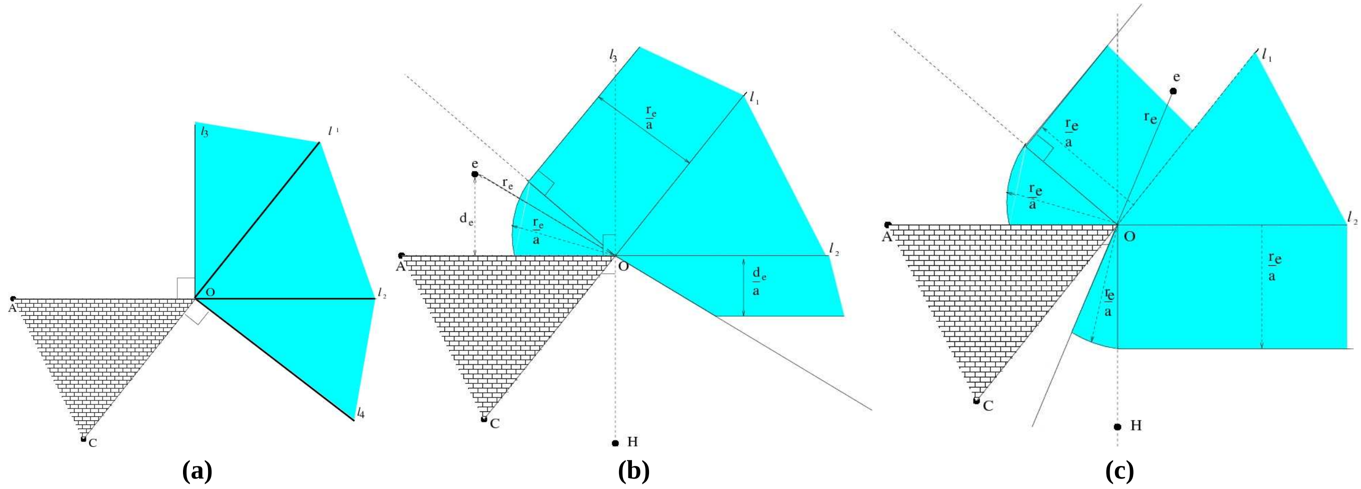
For all points inside the shaded region in figure 6(a), the closest point is the corner. If the closest point to the evader on the obstacle is the corner, to escape to a hidden region on either side of the obstacle as quickly as possible the evader must reach the corner. The evader can escape by following this policy if the pursuer lies in the unshaded region in Figure 6(c), points laying farther than r_e/a from the star region. If the pursuer lies in the star region, no escape is possible.

Using the above idea we present an algorithm in the next section that can decide if the evader can escape for certain initial positions of the pursuer and the evader.

IV. Decidable regions in case of a polygonal environment

In the previous section, we provided a partition of $V(e(t))$ to decide the outcome of the target tracking game. We can conclude that if the pursuer lies outside the shaded region in Figures 6(b) and (c), a strategy exists where evader will

Fig. 6. Evader-based partitions



(a) **(b)** **(c)**

win irrespective of the pursuer policies. The presence of other obstacles does not affect this result.

If the pursuer lies in the shaded region, a strategy, guaranteed to track the evader for all future times and a strategy to do that has been proposed. At this moment, we do not have an extension of this strategy in the presence of multiple obstacles. Hence, we cannot decide the result of the target tracking game if the pursuer starts in the shaded region.

Using the ideas outlined so far in this paper, we propose an algorithm that can decide sufficient conditions for escape of an evader in a general polygonal environment.

Refer to Figures 6(b) and (c). From the previous section, we can conclude that, given a corner, if the pursuer lies outside the region enclosed by rays l_1 and l_2 and the minimum time required by the pursuer to reach rays l_1 or l_2 is greater than the minimum time required by the evader to reach the corner then the evader wins the game. This statement is true even in the presence of other obstacles. To check this condition, compute the shortest distance of the pursuer to the rays l_1 and l_2 and the shortest distance of the evader to the corner. Repeat the process for every corner in the environment. If the condition is satisfied for any corner, the Decidability Algorithm concludes that the evader can escape and the strategy for the evader to escape is to reach that corner along the shortest path. If the condition is not satisfied by any corner, the Decidability Algorithm does not know the result of the target tracking game. The psuedocode of this algorithm is provided in the appendix.

The shortest distance of the evader from the corner can be found by applying Dijkstra's Algorithm to the Visibility Graph of the environment between the initial position of the evader and the corner. To find the shortest distance of the pursuer to rays l_1 and l_2 we construct a *Modified Visibility Graph* (MVG) for a given vertex, v and apply Dijkstra's Algorithm. We present the main steps for the construction of the MVG for a given vertex v.

1) Construct the visibility graph of the environment with edge weights as the Euclidean distance between the two vertices.
2) For every vertex w, check if perpendicular line segments can be drawn from w to rays l_1 and l_2(corresponding to v), without intersecting other obstacles. If only one perpendicular line segment can be drawn to either l_1 or l_2, then compute the length of the perpendicular line segment. If a perpendicular line segment can be drawn to both l_1 and l_2, then compute the length of the smaller perpendicular line segment. Denote the length by d. If the edge wv already exists in the visibility graph, update the weight of the edge to d. Otherwise add an edge with weight d.
3) If for an vertex w, there is no perpendicular line segment to rays l_1 and l_2, do nothing.

The psuedocode for constructing the MVG for a vertex is given in the appendix. The algorithm DECIDABILITYTEST has time complexity of $O(n^3 \log n)$, where n is the number of vertices in the polygonal environment.

V. Extension to One Edge in $\mathbb{R}^3$

Consider an edge in $\mathbb{R}^3$ formed by the intersection of two half planes at an angle of α. Consider a plane π_p perpendicular to both the half planes and passing through the pursuer and a plane π_e perpendicular to both the half planes passing through the evader. Let e_p be the projection of the evader on π_p and p_e the projection of the pursuer on π_e.

Proposition 6 The line $\overline{pe}$ intersects the obstacle iff projected lines $\overline{ep_e}$ and $\overline{pe_p}$ intersect the obstacle.
Proof: Consider any point $(x*, y*, z*)$ on the line $\overline{pe}$. Its projection on π_p, $(x*, y*, z^p)$, lies on $\overline{pe_p}$ and its projection on π_e, $(x*, y*, z^e)$, lies on $\overline{ep_e}$. Since all three points are collinear on a line parallel to the z-axis and the obstacle's shape is invariant along the z-axis if any of the three points intersects the obstacle then the other two will as well. If the visibility line $\overline{pe}$ is broken, then the straight line connecting p to e intersects

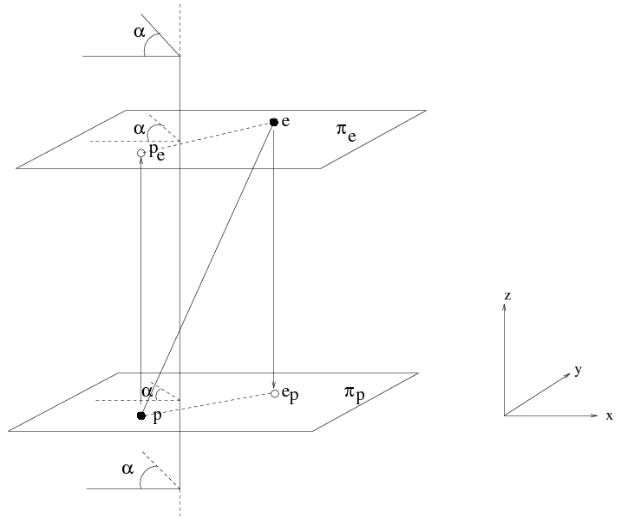


Fig. 7. An edge in three dimensions.

the obstacle at least at one point and the lines $\overline{ep_e}$ and $\overline{pe_p}$ will be broken as well. Otherwise, no point on $\overline{pe}$ is broken and so no point on $\overline{ep_e}$ and $\overline{pe_p}$ will be broken. ■

Proposition 5 shows that, in this particular geometry, the problem in $\mathbb{R}^3$ can be reduced to to a problem in $\mathbb{R}^2$. Given the maximum velocities of the pursuer and evader, the decomposition of the visible space is simply the extrusion of the planar decomposition of the visibility region along the z axis.

VI. Conclusions

This paper addresses the problem of surveillance in an environment with obstacles. We show that the problem of tracking an evader with one pursuer around one corner is completely decidable. We present a partition of the visibility region of the pursuer where based on the region in which the evader lies, we provide strategies for the evader to escape the visibility region of the pursuer or for the pursuer to track the target for all future time. We also present the solution to the inverse problem: given the position of the evader, the positions of the pursuer for which the evader can escape the visibility region of the target. These results have been provided for varying speeds of the pursuer and the evader. Based on the results of the inverse problem we provide an $O(n^3 \log n)$ algorithm that can decide if the evader can escape from the visibility region of a pursuer for some initial pursuer and evader positions. Finally, we extend the result of the target tracking problem around a corner in two dimensions to an edge in three dimensions. We have shown that the target tracking game in $\mathbb{R}^3$ can be reduced to a target tracking game in $\mathbb{R}^2$

In our future work, we plan to address the decidability of the target tracking problem in general polygonal environment. We are using game theory as a tool to increase the decidable regions for the problem. We also plan to use tools in topology and computational geometry to understand the problem to a greater depth. Finally, we would like to extend the problem to multiple pursuers and multiple evaders and use control theory to provide strategies for successful tracking.

References

[1] S. Alexander, R. Bishop, and R. Ghrist. Pursuit and evasion in non-convex domains of arbitrary dimensions. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[2] T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory, 2nd Ed.* Academic, London, 1995.

[3] T. Bandyopadhyay, Y. Li, M.H. Ang Jr, and D. Hsu. Stealth Tracking of an Unpredictable Target among Obstacles. *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics*, 2004.

[4] T. Bandyopadhyay, Y. Li, M.H. Ang Jr, and D. Hsu. A greedy strategy for tracking a locally predictable target among obstacles. *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics*, 2006.

[5] C. Becker, H. Gonzalez-Banos, J.C. Latombe, and C. Tomasi. An intelligent observer. *Proceedings of International Symposium on Experimental Robotics*, pages 94–99, 1995.

[6] H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. *Principles of robot motion: theory, algorithms, and implementations (Intelligent robotics and autonomous agents)*. The MIT Press, Cambridge, MA, 2005.

[7] A. Efrat, HH Gonzalez-Banos, SG Kobourov, and L. Palaniappan. Optimal strategies to track and capture a predictable target. *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, 3, 2003.

[8] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *Robotics and Automation, IEEE Transactions on*, 8(3):313–326, 1992.

[9] P. Fabiani and J.C. Latombe. Tracking a partially predictable object with uncertainty and visibility constraints: a game-theoretic approach. Technical report, Technical report, Univeristy of Stanford, December 1998. http://underdog. stanford. edu/.(cited on page 76).

[10] HH Gonzalez-Banos, C.Y. Lee, and J.C. Latombe. Real-time combinatorial tracking of a target moving unpredictably among obstacles. *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, 2, 2002.

[11] Leonidas J. Guibas, Jean-Claude Latombe, Steven M. LaValle, David Lin, and Rajeev Motwani. A visibility-based pursuit-evasion problem. *International Journal of Computational Geometry and Applications*, 9(4/5):471–, 1999.

[12] L. Guilamo, B. Tovar, and S. M. LaValle. Pursuit-evasion in an unknown environment using gap navigation trees. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[13] JP Hespanha, M. Prandini, and S. Sastry. Probabilistic pursuit-evasion games: a one-step Nash approach. *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, 3, 2000.

[14] S. A. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics & Automation*, 12(5):651–670, October 1996.

[15] R. Isaacs. *Differential Games*. Wiley, New York, 1965.

[16] V. Isler, S. Kannan, and S. Khanna. Locating and capturing an evader in a polygonal environment. *Workshop on Algorithmic Foundations of Robotics (WAFR04)*, 2004.

[17] V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion in a polygonal environment. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 21(5):875–884, 2005.

[18] B. Jung and G.S. Sukhatme. Tracking Targets Using Multiple Robots: The Effect of Environment Occlusion. *Autonomous Robots*, 13(3):191–205, 2002.

[19] Swastik Kopparty and Chinya V. Ravishankar. A framework for pursuit evasion games in rn. *Inf. Process. Lett.*, 96(3):114–122, 2005.

[20] Parker L. Algorithms for Multi-Robot Observation of Multiple Targets. *Journal on Autonomous Robots*, 12:231–255, 2002.

[21] S. M. LaValle. *A Game-Theoretic Framework for Robot Motion Planning*. PhD thesis, University of Illinois, Urbana, IL, July 1995.

[22] S. M. LaValle, H. H. Gonzalez-Banos, C. Becker, and J. C. Latombe. Motion strategies for maintaining visibility of a moving target. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 731–736, Albuquerque, NM, USA, April 1997.

[23] 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, April 2001.

[24] S. M. LaValle, D. Lin, L. J. Guibas, J.-C. Latombe, and R. Motwani. Finding an unpredictable target in a workspace with obstacles. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 737–742, 1997.

[25] S. M. LaValle, B. Simov, and G. Slutzki. An algorithm for searching a polygonal region with a flashlight. *International Journal of Computational Geometry and Applications*, 12(1-2):87–113, 2002.

[26] E. Malis, F. Chaumette, and S. Boudet. 2D 1/2 Visual Servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, 1999.

[27] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust realtime visual tracking using a 2d-3d model-based approach. *IEEE Int. Conf. on Computer Vision, ICCV99*, 1:262–268.

[28] T. Muppirala, S. Hutchinson, and R. Murrieta-Cid. Optimal Motion Strategies Based on Critical Events to Maintain Visibility of a Moving Target. *Robotics and Automation, 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3826–3831, 2005.

[29] R. Murrieta, A. Sarmiento, S. Bhattacharya, and SA Hutchinson. Maintaining visibility of a moving target at a fixed distance: the case of observer bounded speed. *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, 1.

[30] R. Murrieta, A. Sarmiento, and S. Hutchinson. On the existence of a strategy to maintain a moving target within the sensing range of an observer reacting with delay. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2, 2003.

[31] R. Murrieta-Cid, H. H. Gonzalez-Banos, and B. Tovar. A reactive motion planner to maintain visibility of unpredictable targets. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 4, pages 4242–4248, 2002.

[32] R. Murrieta-Cid, T. Muppirala, A. Sarmiento, S. Bhattacharya, and S. Hutchinson. Surveillance strategies for a pursuer with finite sensor range. *Robotics Research, 2007. International Journal on*, 2007. To Appear.

[33] TD Parsons. The search number of a connected graph. *9th Southeastern Conference on Combinatorics, Graph Theory and Computing*, pages 549–554.

[34] TD Parsons. Pursuit-evasion in a graph. *Theor. Appl. Graphs, Proc. Kalamazoo*, 1976.

[35] L. A. Petrosjan. *Differential Games of Pursuit*. World Scientific, Singapore, 1993.

[36] PE Rybski, NP Papanikolopoulos, SA Stoeter, DG Krantz, KB Yesin, M. Gini, R. Voyles, DF Hougen, B. Nelson, and MD Erickson. Enlisting rangers and scouts for reconnaissance and surveillance. *Robotics & Automation Magazine, IEEE*, 7(4):14–24, 2000.

[37] P.E. Rybski, S.A. Stoeter, M.D. Erickson, M. Gini, D.F. Hougen, and N. Papanikolopoulos. A team of robotic agents for surveillance. *Proceedings of the fourth international conference on autonomous agents*, pages 9–16, 2000.

[38] S. Sachs, S. Rajko, and S. M. LaValle. Visibility-based pursuit-evasion in an unknown planar environment. *International Journal of Robotics Research*, 23(1):3–26, January 2004.

[39] I. Suzuki and M. Yamashita. Searching for a Mobile Intruder in a Polygonal Region. *SIAM Journal on Computing*, 21:863, 1992.

[40] B. Tovar and S. M. LaValle. Visibilty-based pursuit-evasion with bounded speed. In *Proceedings Workshop on Algorithmic Foundations of Robotics*, 2006.

[41] R. Vidal, O. Shakernia, HJ Kim, DH Shim, and S. Sastry. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *Robotics and Automation, IEEE Transactions on*, 18(5):662–669, 2002.

APPENDIX

The subroutine VG(S), computes the visibility graph of a polygonal environment S.

Algorithm DECIDABILITYTEST(S,P,E,v_p,v_e)
Input: A set S of disjoint polygonal obstacles, The Pursuer position P, The evader position E, Pursuer maximum velocity v_p, Evader maximum velocity v_e
Output: result: If Evader wins YES, else DONOTKNOW

```
$G_{visE}$ =VG(S $\cup$ E)
$G_{mvisP}$ =MVG(S $\cup$ P)
i$\leftarrow$1
Until $i = n + 1$ or $flag = false$
        $l_1$ =DIJKSTRA(VG(S),E,$V_i$)
        $l_2$ =DIJKSTRA(MVGCONSTRUCT(S,$V_k$),P,$V_i$)
        if $\frac{l_2}{v_p} > \frac{l_1}{v_e}$
            result=WINS
        else $i = i + 1$
if $i = n + 1$, result=DONOT KNOW
```

In the psuedocode below, the subroutine VISIBLEVERTICES(v,S) is an algorithm which provides the vertices of a polygonal environment S, visible to a vertex v. It can be implemented using a rotational plane sweep algorithm [6].

Algorithm MVGCONSTRUCT(S,V_k)
Input: A set S of disjoint polygonal obstacles, A vertex V_k
Output: The Modified Visibility Graph $G_{MVIS}(S)$

```
Initialize a graph $G = (V, E)$ where V is the set of all
vertices of the polygons in $S$ and E= $\emptyset$
for all vertices $v \in V$
    do W $\leftarrow$ VISIBLEVERTICES(v,S)
        for every vertex $w \in W$ and $w \neq V_k$
            add edge $(v, w)$ to E with weight as the
            euclidean length of segment $\overline{vw}$.
for all vertices $v \in$ G
    d$\leftarrow$MINLSDIST(v,$V_k$)
        If d$\neq \infty$
            if edge v$V_k$ exists, update weight of v$V_k$
            to d
        else add edge v$V_k$ to E with weight d
```

In the following psuedocode the subroutine

1) ORTHOGONALITYCHECK(v,k) is to check if v lies outside rays l_1 and l_2 and if we can draw a perpendicular from v to rays l_1 and l_2 without intersecting other obstacles. It returns TRUE if either of the conditions is satisfied
2) CLOSER(v, l_1, l_2) is used to find which ray among l_1 and l_2 has a lesser distance from v.
3) MINDISTANCE(v,k) is to compute the minimum distance from v to a ray k.

Algorithm MINLSDISTANCE(v, V_k)
Input: A vertex v in G, a vertex V_k in G
Output: d:the length of the line perpendicular from v to l_1 and l_2 corresponding to V_k. If the perpendicular line intersects any obstacles then it gives an output of ∞.

```
If ORTHOGONALITYCHECK(v,$l_1, l_2$)=TRUE
        k=CLOSER$(v, l_1, l_2)$
        If COLLISIONCHECK(v,k)=TRUE
            d$\leftarrow \infty$
        else d$\leftarrow$MINDISTANCE(v,k)
else d$\leftarrow \infty$
```