Visibility-Based Pursuit-Evasion with Bounded Speed

Benjamín Tovar and Steven M. LaValle

Dept. of Computer Science
University of Illinois at Urbana-Champaign
61801, USA
{btovar,lavalle}@uiuc.edu

Summary. This paper presents an algorithm for a visibility-based pursuit-evasion problem in which bounds on the speeds of the pursuer and evader are given. The pursuer tries to find the evader inside of a simply-connected polygonal environment, and the evader in turn tries actively to avoid detection. The algorithm is at least as powerful as the complete algorithm for the unbounded speed case, and with the knowledge of speed bounds, generates solutions for environments that were previously unsolvable. Furthermore, the paper develops a characterization of the set of possible evader positions as a function of time. This characterization is more complex than in the unbound-speed case, because it no longer depends only on the combinatorial changes in the visibility region of the pursuer.

1 Introduction

Consider a robot in a search-and-rescue operation, such as firefighting inside a building. Victims have to be located before they can receive proper aid. The objective of this robot, called here the *pursuer*, is to find each person inside the building. In the worst-case, the robot should plan as if a person, called an *evader*, is actively hiding. However, the pursuer can make some safe assumptions about each evader. For example, a person does not move at more than $12m/s$. This paper studies the search taking into consideration such speed bounds.

We consider a version of the visibility-based pursuit-evasion problem, in which bounds on the speed of the pursuer, and the evader are given. This yields a major complication for describing the set of possible positions where the evader might be. Perhaps surprisingly, describing the possible positions of the evader with unbounded speed is much easier; they depend only on the combinatorial changes in the visibility region of the pursuer. This is no longer true in the bounded speed case, because the set of possible evader positions is also a function of *time*.

Determining the set of possible evader's positions as a function of time, called the *reachable set of the evader* has been previously studied in [2, 7, 14]. Even in the absence of obstacles, the exact computation of the reachable set is computational intractable, since it involves finding a solution to the Hamilton-Jacobi-Bellman equation [6]. The present paper is the first attempt to describe the set of evader's position inside a polygonal environment. Whereas this description is

S. Akella et al. (Eds.): Algorithmic Foundation of Robotics VII, STAR 47, pp. 475–489, 2008.
springerlink.com

made exact, it is rather used to prove that an approximation to the reachable set that is easier to compute is conservative. This approximation is then used together with the combinatorial changes in the visibility of the robot, enlarging the class of environments that can be searched by a single pursuer. Thus, even though knowing speed bounds makes the problem *easier* to the pursuer, since evader capabilities are decreased, the design of a complete algorithm becomes much more complicated. This is one of the reasons of why the speed bounds have been ignored for visibility-based pursuit-evasion.

The visibility-based pursuit-evasion problem was proposed in [15]. The unbounded speed case has been discussed extensively in the literature. A complete algorithm for a pursuer with an omnidirectional field of view was presented in [4]. A solution for a limited pursuer's field of view was presented in [3]. For pursuers moving on the boundary of the environment, having a single ray of visibility, a complete algorithm was presented in [12]. For the same problem, an finite state automaton was designed in [10]. A randomized solution for a pursuer moving under polyhedral kinematic constraints was described in [8], based on a randomized strategy presented in [9]. The randomized algorithm gives an arbitrarily high probability of evader detection, even when the environment is not searchable with one pursuer by the complete algorithm in [4]. Minimal sensing solutions, in which the environment is unknown to the pursuer, have been presented in [5, 13].

This paper formalizes the problem of pursuit-evasion with bounded speed. We give a description of the set of evader possible positions, *contaminated regions*, in the form of an information state. This information state takes advantage of the combinatorial structure studied in previous approaches to compute the worst-case contamination of a region. Contaminated regions are not kept explicitly, but are computed selectively as the pursuer needs them. Assuming a pursuer that moves in piecewise-linear paths, we present a search algorithm that uses the description of the contaminated regions as a function of the evader speed. This algorithm is as powerful as any complete algorithm for the unbounded speed case. The movement of the pursuer presents a challenging optimization problem [7, 17, 18, 19]; thus, moving the pursuer in piecewise-linear paths may not lead to a complete algorithm. However, by taking into account the speed bounds defined in the problem, this algorithm solves many instances of pursuit-evasion tasks in environments for which no solution exists in the unbounded speed case.

2 Problem Formulation

The pursuer and the evader are modeled as points moving in an open set $\mathcal{R} \subset \mathbb{R}^2$. It is assumed that $\mathcal{R}$ is simply-connected, with a polygonal boundary $\partial\mathcal{R}$. Let $e(t) \in \mathcal{R}$ denote the position of the evader at time $t \geq 0$. It is assumed that $e : [0, \infty) \to \mathcal{R}$ is a continuous function. Let $\mathcal{V}_e(t)$ be the speed of the pursuer at time t. The mapping $\mathcal{V}_e : [0, \infty) \to [0, v_e]$ may not be continuous, but sets a maximum speed for the evader at v_e. Similarly, let $p(t) \in \mathcal{R}$ denote the position of the pursuer at time $t \geq 0$. It is assumed that $p : [0, \infty) \to \mathcal{R}$ is continuous

and piecewise-differentiable. The pursuer moves with a maximum speed of v_p according to the speed map $\mathcal{V}_p : [0, \infty) \to [0, v_p]$, which may not be continuous. Since dynamics is disregarded, the implication of the speed bounds are better understood if the position mappings are parametrized as a function of their arclength s. For example, if the length of the path from $e(t_0)$ to $e(t_f)$ is s, then $s \leq v_e(t_f - t_0)$, for any $t_0, t_f \in [0, \infty)$, $t_0 \leq t_f$.

For a point $q \in \mathcal{R}$, let $V(q)$ denote the set of all points in $\mathcal{R}$ that are visible from q (i.e., the line segment joining q and any point in $V(q)$ lies in $\mathcal{R}$). The set $V(q)$ is called the visibility region at q. A mapping $p(t)$ is called a solution strategy if for every continuous path $e : [0, \infty) \to \mathcal{R}$ subject to $\text{arclength}(e(t_0), e(t)) \leq v_e(t - t_0)$, $\forall t_0, t \in [0, \infty)$, there exists a time $t_c \in [0, \infty)$ such that $e(t_c) \in V(p(t_c))$. The time t_c is called the time of capture for the strategy $p(t)$. Thus, the position of the evader remains unknown to the pursuer until t_c. The pursuer's task is to find a $p(t)$ solution strategy with a finite time of capture. A complete algorithm reports such a solution strategy if it exits, or reports that evader remains undetected for the given speed bounds.

It is clear that the particular values of speeds of the pursuer and evader are not as important as the ratio v_e/v_p between them. For each simple polygon and each evader speed, a pursuer speed can be found such that a solution strategy exists:

Proposition 1. *Given a simply-connected polygonal environment $\mathcal{R}$ and a maximum evader speed v_e, a speed of a pursuer v_p can be found such that a solution strategy exists.*

Proof. Compute the visibility graph of $\mathcal{R}$ and find the edge with the smallest length l_{min}. Set $v_p = l_{\mathcal{R}} v_e / l_{min}$, in which $l_{\mathcal{R}}$ is the length of $\partial\mathcal{R}$. If the pursuer transverses $\partial\mathcal{R}$ at such speed, any evader is detected. This is because the evader can only hide from reflex vertex to reflex vertex (bitangents), but the pursuer sees all such paths before the evader can transverse them. □

Proposition 1 motivates the study of visibility pursuit-evasion with bounds on the speed. Rather than declaring the problem unsolvable, bounds on the speed may be found such that a strategy exists. An interesting question considers finding the maximum v_e/v_p for which there exists a solution. If the maximum v_e/v_p approaches infinity, then $\mathcal{R}$ can be searched assuming an evader with unbounded speed. Likewise, if it approaches 0, then the search looks like a visibility coverage problem. An upper bound for this ratio can be found with a binary search using the strategy presented in this paper.

2.1 The Model

The pursuer has perfect information about its position and orientation with respect to $\mathcal{R}$. It has two sensors, a clock and a visibility sensor. The clock reports a positive real number that indicates the time elapsed from the beginning of the pursuing task. The visibility sensor $V : \mathcal{R} \to \text{pow}(\mathcal{R})$ reports the visibility region from the current position of the pursuer. An observation space is defined

as $Y = [0, \infty) \times \text{pow}(\mathcal{R})$. It can be interpreted as visibility regions together with a timestamp. Let $\tilde{y}_t$ be the history (sequence) of observations up to time t, and let $\tilde{p}_t$ be the history of all the pursuer positions up to time t. Also, let $x = (p(t), e(t))$ be the *state* of the pursuit-evasion task. This leads to a state space $X = \mathbb{R}^2 \times \mathbb{R}^2 = \mathbb{R}^4$. Consider the history information state $(\eta_0, \tilde{p}_t, \tilde{y}_t)$, in which the initial condition $\eta_0 = (p(0), \mathcal{R})$ reflects the fact that at $t = 0$ the position of the pursuer is known, but the evader can be anywhere in $\mathcal{R}$. Let $X(\eta_0, \tilde{p}_t, \tilde{y}_t) \subseteq X$ be the smallest set of states in which the pursuit-evasion task might be, as it is deduced from $(\eta_0, \tilde{p}_t, \tilde{y}_t)$. Each $\eta_t = X(\eta_0, \tilde{p}_t, \tilde{y}_t)$ is an information state of the nondeterministic information space $\mathcal{I}_{ndet} = \text{pow}(X)$ (see [11]). The information state η_t is represented as $\eta_t = (p(t), E(t))$, in which the set $E(t) \subset \mathcal{R}$ is the set of all positions in which the evader might be at time t. Consider the maximal connected sets of points in $E(t)$. Each of these sets is referred to as a *contaminated* region of $\mathcal{R}$. Each of the maximal connected sets of points in $\mathcal{R} \backslash E(t)$ is referred to as a *cleared* region of $\mathcal{R}$. When a contaminated region becomes cleared and later it becomes contaminated again, such region is referred to as *recontaminated*. An equivalent way to describe a solution strategy is to find the mapping $p(t)$ such that the state (p, e) is known. In other words, the task is completed when $E(t)$ contains a single point (the location of the evader), or it is the null set (no evader is in $\mathcal{R}$).

Let $y_{t'} \in Y$ be the observation made at time t'. The information transition equation is defined as $\eta_{t'} = f_{\mathcal{I}}(\eta_t, p(t'), y_{t'})$. The main complication of determining $f_{\mathcal{I}}$ lies in describing how $E(t)$ changes. The next section describes the changes in $E(t)$ as a function of time and the pursuer movements. This description is used later to find a solution strategy.

3 Describing Contaminations

The edges of a visibility polygon $V(p)$ alternate between being part of $\partial\mathcal{R}$ or crossing the interior of $\mathcal{R}$. The latter ones are collinear with p, and are referred to as *gaps*. A label of *contaminated* or of *cleared* is assigned to each gap. The label indicates whether the maximal connected region in $\mathcal{R} \backslash V(p)$ for which the gap is an edge might contain the evader. As the pursuer moves, $V(p)$ changes combinatorially. For the unbounded speed case, $E(t)$ depends uniquely on the gaps changes. If gap α appears, the region behind α is cleared, and α is labeled accordingly. If a cleared gap α merges with a contaminated gap β to form gap γ, then the whole region behind α gets recontaminated, and γ is labeled as contaminated. Gap changes occur when the pursuer crosses inflection rays (appearances and disappearances), or when it crosses bitangent complements (merges and splits). For a deeper discussion of gap changes the reader is referred to [16].

These conditions no longer hold when bounds on the speed are present. A gap does not have to disappear and appear again to mark the whole region behind it as cleared. Likewise, when two gaps merge, a cleared region does not contaminate immediately. Before modeling contaminations, we introduce some

gap terminology. A gap that can disappear is called *primitive*. Otherwise, if it can split, it is called *nonprimitive*. For a gap α starting at reflex vertex a_α, let l_v and l_n be the edges of $\partial\mathcal{R}$ that intersect at a_α. As shown in Figure 1, l_v is the edge that is (perhaps partially) visible from p, while l_n is completely hidden. Let $\theta(t)$ be the angle between α and l_n at time t. If $\theta(t)$ increases, α is said to move in the *positive direction*. The length of α at time t is denoted by $\lambda(t)$. Let i_α and i_n be the ray extensions of α and l_n respectively, until an edge of $\partial\mathcal{R}$ is hit. If α is primitive, i_n is the inflection ray that, if crossed, forces α to disappear. Also, let $\alpha(r, \theta(t))$ be the point on α at r distance from a_α when α is at the angular position $\theta(t)$. Finally, let $b(t) = \alpha(\lambda(t), \theta(t))$ be the intersection point of α with $\partial\mathcal{R}$. Thus, α is the line segment $[a_\alpha, b(t)]$.

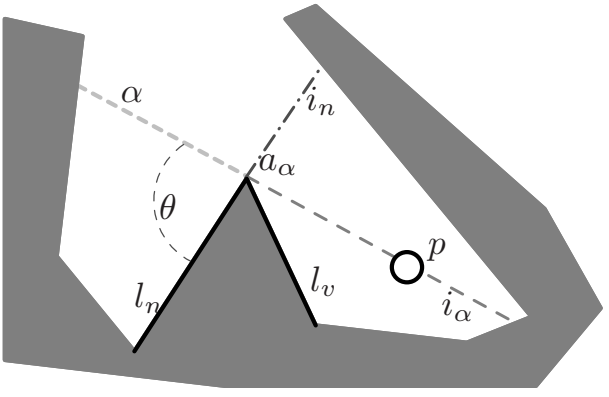

Fig. 1. Gap description. The gap α starts at reflex vertex a_α. It has an angular position of θ, as measured from the edge l_n. The rays i_α, and i_n, extend from a_α in the direction of α and l_n respectively, until they hit an edge of the polygon.

3.1 A Recontamination Fan

Let α be a gap currently visible for which the region behind it is completely contaminated at time $t = 0$. Assume that from $t = 0$ to $t = t_f$, α does not disappear, split or merge. The region between α at $\theta(0)$, and α at $\theta(t_f)$ is called the *recontamination fan* of α, or α*-fan*. Recontamination inside the α-fan is described next. If α moved in the negative direction, that is $\theta(0) > \theta(t_f)$, the whole region behind α is still contaminated and no more computations are needed. For the positive direction the angular velocity $\omega(t)$ of α is needed. Assume that the vertex a_α is placed at the origin of the plane, and that the pursuer position is given by $p = (p_x(t), p_y(t))$. Let $\mathbf{p}$ be the position vector of p, and let $\mathbf{n}$ be its unit normal vector. Also, let $\mathbf{v_p} = [\mathrm{d}p_x/\mathrm{d}t, \mathrm{d}p_y/\mathrm{d}t]$ be the velocity of the pursuer. We have:

$$\omega(t) = \frac{\mathbf{v_p} \cdot \mathbf{n}}{|\mathbf{p}|} = \frac{p_x \frac{\mathrm{d}y}{\mathrm{d}t} - p_y \frac{\mathrm{d}x}{\mathrm{d}t}}{p_x^2 + p_y^2} \tag{1}$$

Assume that $\omega(t)$ has a single maximal value at $\omega(t_{max}) = \omega_{max}$. The general case, when $\omega(t)$ has several critical points is discussed later in this section.

Recontamination from $t = 0$ to $t = t_{max}$

Consider an evader arbitrarily close to $\alpha(r, \theta(0))$. If the evader is to remain arbitrarily close to $\alpha(r, \theta(t))$ during $t \in [0, t_{max}]$, then $\omega_{max} \leq v_e/r$. Let $r_n = v_e/\omega_{max}$. Any evader at distance less than or equal to r_n from a_α can follow exactly the angular motion described by α. Thus, the evader could be anywhere along this arc and such region remains contaminated (see Figure 2.a). Consider now $r_b = v_e/\omega(0)$. Any evader arbitrarily close to $\alpha(r_b, \theta(0))$ cannot follow the angular motion described by α. This is true for positions on the gap with $r \geq r_b$. Their recontamination regions can be described with a circular section with radius $v_e t_{max}$, centered at the original position of the evader, as shown in Figure 2.b. The circular sections do not grow towards a_α. This is because either the positions reachable are already considered by a radius of smaller length, or because the evader would otherwise cross α. The latter is better exemplify if $w(t)$ remains constant at $w(0)$. In this case, r_n and r_b coincide. The region after the arc at r_n is bounded by the involute of a circle (see Figure 2.b). These circular sections cannot intersect α at any time before t_{max}, since $w(t)$ never decreases. Adding all the circular sections for radii larger than r_b makes the portion of α starting at r_b and ending at $b(0)$ sweep perpendicularly by $v_e t_{max}$ (see Figure 2.b). As α moves, its length $\lambda(t)$ may change. If $\lambda(t)$ decreases, the $v_e t_{max}$ sweep is done as described, but eliminating the part that intersects with the polygon. If $\lambda(t)$ increases, a contamination region grows as a circular section, with radius $v_e t_{max}$ and center $b(0)$. This circular section is also present when $r_n > \lambda(0)$. In this case, the region of the circular section crossing (above) α is eliminated.

For $r \in (r_n, r_b)$, let t_r be the smallest time for which $v_e \geq w(t)r$ does not hold. This time is referred to as the *breaking time* of r. Before t_r, an evader placed arbitrarily close to $\alpha(r, \theta(t))$ follows the arc described by α. From t_r to $t_{\max}$, the evader could be anywhere in a growing circular section centered at the last point of contact with α at $\theta(t_r)$, and bounded by α at $\theta(t)$. When adding the effect of such contamination regions, a sweep similar to the one described before takes place. The difference is that, per radius r, the sweep length is $v_e(t_{max} - t_r)$ and the evader travels perpendicularly to α at $\theta(t_r)$.

Recontamination from $t = t_{max}$ to $t = t_f$

Since $w(t)$ is now decreasing, positions for which a breaking time existed may be able to intersect α in the sweep $v_e(t_f - t_{max})$. Suppose the sweep intersects α at $\alpha(r_m, \theta(t_f))$, and r_m is the maximum radius for all such intersections. Then the circular section with radius r_m and center a_α, bounded by α at $\theta(0)$ and $\theta(t_f)$, is completely contaminated. This circular section is added to the contamination sweep that did not intersect α, namely the sweep of the radii after r_m. In general, the contamination boundary consists of:

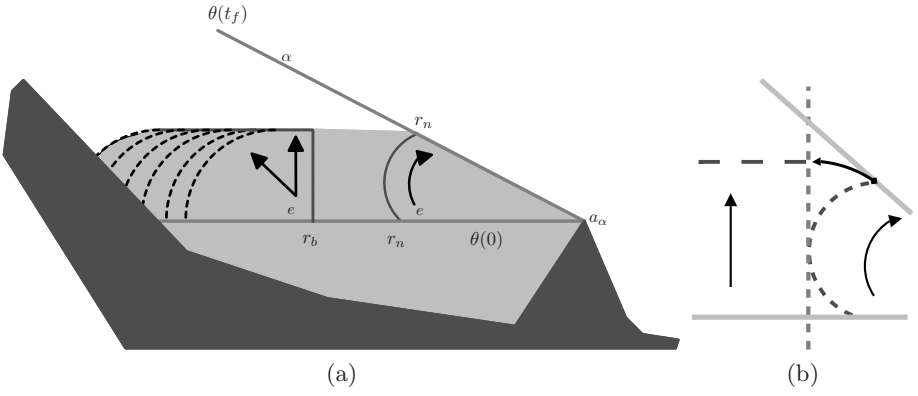

Fig. 2. Recontamination fan. (a) Gap α moved from $\theta(0)$ to $\theta(t_f)$. Any evader before r_n follows exactly the angular motion of α. Any evader after r_b can be anywhere inside a circular section of radius $v_e t_f$. The net effect is a line segment sweeping perpendicularly to the original position. (b) For $w(t)$ constant, the contamination region between the vertical dashed line and the dashed arc is bounded by the involute of a circle.

1. A line segment from a_α to $\alpha(r_m, \theta(t_f))$
2. A curve function of $p(t)$ from $\alpha(r_m, \theta(t_f))$ to the point $(r_b, v_e t_f)$,
3. A line segment $[(r_b, v_e t_f), (b(0), v_e t_f)]$, parallel to α at $\theta(0)$.
4. A circular section, with center $b(0)$ and radius $v_e t_f$.

For some values of r_m some elements may not be present. Note that the region between α at $\theta(t_f)$ and the contamination boundary is cleared, In the unbounded speed case, the whole region behind α would be marked as contaminated. When $w(t)$ has more than one maximum, the concepts before described are applied as follows. Find all the local minima of $\omega(t)$ in $[0, t_f]$. Assume the local minima occur at $t_1, t_2, ..., t_n$. The recontamination fan is computed by parts, from $t = 0$ to t_1, from the gap at $\theta(0)$ to $\theta(t_1)$, and so on. At each step, new breaking times should be computed, and the sweep of $v_e t$ should be checked for intersection. Note that the contamination boundary may contain two or more elements of the same type. This is because the breaking times would change for each radius at each period of time. Particularly, we can consider the line segment up to r_m of the previous time period as a gap, for which a recontamination fan is computed.

Piecewise-Linear Approximation

While a complete algorithm should compute the contamination boundary exactly, a piecewise linear approximation is easily computed. Let $r_{n'} = v_e t_f / \tan(\theta(t_f) - \theta(0))$. An evader traveling $v_e t_f$ from $\alpha(r_{n'}, \theta(0))$ and perpendicularly to α at $\theta(0)$, intersects α at $\alpha(r_{m'}, \theta(t_f))$, with $r_{m'}^2 = r_{n'}^2 + (v_e t)^2$. The approximation is a line segment from a_α to $\alpha(r_{m'}, \theta(t_f))$, and a line segment $\alpha(r_{m'}, \theta(t_f))$ parallel to α at $\theta(0)$ that extends until an edge of $\partial\mathcal{R}$ is hit (see Figure 3). Note

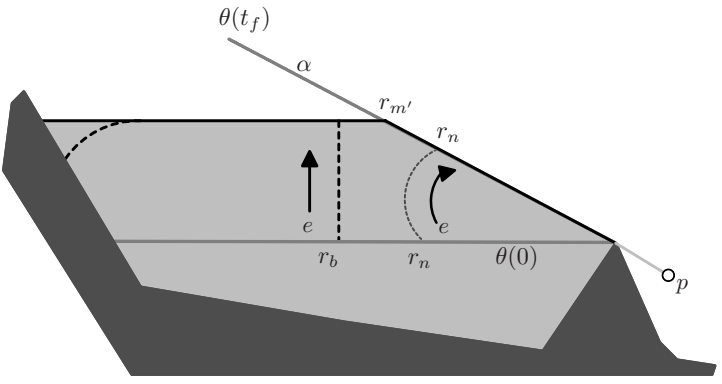

Fig. 3. Piecewise-linear approximation. Instead of computing the exact contamination boundary, an approximation is easily generated by ignoring the angular velocity of the gap, considering only its final angular position.

that $r_{m'} > r_m$, because traveling from $r_{m'}$ did not have to wait for a breaking time to intersect α. Thus, this approximation is conservative and may be preferred over the exact one in real robotic implementations given its simplicity.

Fan Contamination for a Pursuer Moving in a Piecewise-Linear Path

As an example of a fan recontamination, consider a pursuer that moves in a piecewise-linear path with constant velocity v_p. To simplify the example, assume that the pursuer moves in a vertical line at a distance x_0 of the y-axis. From Equation 1:

$$\omega(t) = \frac{x_0 v_p}{x_0^2 + p_y^2} = \frac{x_0 v_p}{x_0^2 + (tv_p - y_0)^2} \tag{2}$$

Equation 2 has a maximum at $\omega(y_0/v_p) = v_p/x_0$. Thus, $r_n = v_e x_0/v_p$, the last radius for which an evader can follow exactly the angular motion of α. Based on Equation 2, an expression for the breaking time of each radius can be found using $\omega(t)r = v_e$:

$$t(r) = \frac{1}{v_p}\left(y_0 + \sqrt{\frac{r x_o v_p}{v_e} - x_o^2}\right) \tag{3}$$

Note that $t(r)$ for $r < r_n$ generates complex solutions. This means, as expected, that such breaking times do not exist. Figure 4 shows a computed example of the fan recontamination for two different speeds of the evader.

3.2 Merges Spreading Contamination

When two gaps merge, contamination spreads in the regions behind them. For two or more consecutive reflex vertices in $\partial\mathcal{R}$, a merge is also considered when one of them occludes the others. While this is not entirely true, since a bitangent

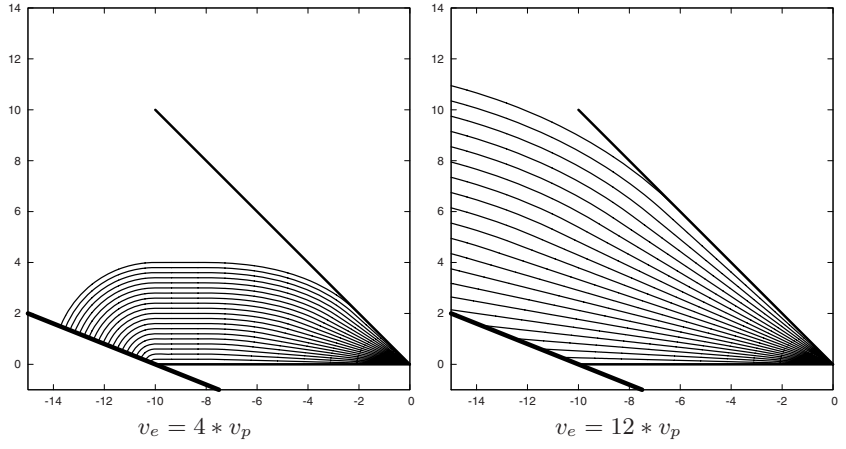

Fig. 4. Recontamination fan computed example for a pursuer linear motion. The gap start at $\theta(0) = 0°$, and ends at $\theta(1) = 45°$. The thick line on the bottom-right represents an edge of $\mathcal{R}$. The evolution of the contamination is shown for times between $[0, 1]$.

does not exists, it simplifies the description of contamination. Particularly, a primitive gap is assigned to the first occluded vertex, while a nonprimitive gap is assigned to each of the remaining consecutive reflex vertices. The *split* only generates one gap. To describe the recontamination between gaps, the following lemma is proposed:

Lemma 1. *Let α and β be two gaps that merge into gap γ. When γ splits, α and β appear at the same angular position at the time of the merge, independently from the pursuer motion.*

Proof. Merges and splits occur when the pursuer crosses a bitangent complement of $\partial\mathcal{R}$. Thus α, β, and γ are aligned with the bitangent at the split or the merge. This is independent from where the bitangent complement is crossed. □

Lemma 1 provides a tool to encode the contamination of cleared regions. If α is a gap for which the region behind is completely cleared, the angular position of α before a merge (i.e., when it was last seen) is recorded. The merge may allow a path between α and a contaminated gap β that the evader can transverse without being detected. Assume that the merge occurs at time $t = 0$, and contamination should be determined for time t_f. Further, assume that γ splits at time $t_s \in [t_0, t_f]$. If the evader does not cross γ by the time t_s, it cannot contaminate α anymore. The worst-case contamination of α has two general cases, function of whether α is visible from the position of the evader or not.

First, assume that α is completely visible from the current position of the evader, which is at h distance from a_α. For clarity, we disregard the effect of the recontamination fans for now. The evader should move as to maximize the

contamination of the region behind α. If $v_e t_f > h$, then the worst-case appears when the evader moves to a_α. This forces the pursuer to cross the inflection ray i_n to completely clear the region again. If $v_e t_f \leq h$, the evader cannot reach a_α, but recontamination may still exist. Imagine the evader moves towards some point in α. When the evader reaches this point and keeps moving, allow α and its extension ray i_α to move with the evader around a_α. We said that the evader is *pushing* the gap. When the evader reaches a point in l_n, the rays i_n and i_α coincide (see Figure 5). If the pursuer crosses i_α before the evader gets to l_n, the evader is detected. Since at the moment of the merge i_α is collinear with the pursuer, the worst-case pushes i_α as far as possible from the pursuer. Thus, in the worst-case, the evader pushes α as to minimize $\theta(t_f)$, as any other movement would make its detection easier. The following lemma provides the optimal movement for the evader:

Lemma 2. *Let α be a cleared gap, let $l = v_e t_f$, and consider an evader standing at distance h from a_α, with $t = 0$ and α completely visible from the evader position. If $l > h$, then the optimal strategy for the evader is to move to a_α. If $l \leq h$, then it should move in a straight line, with an angle of $\phi = \arccos(l/h)$ as measured from a parallel line to α, passing through the position of the evader.*

Proof. When $l \geq h$, moving to a_α makes i_α coincide with i_n. When $l < h$, assume the optimal strategy is not straight line. Such strategy has an endpoint b, which can be joined with the position of the evader by a line segment, reaching b faster, a contradiction. To find the angle ϕ, consider the angle $\sigma = \theta(0) - \theta(t_f)$, in which $\theta(0)$ and $\theta(t_f)$ are the positions of the gap before the evader pushed it. To maximize σ, consider the triangle (see Figure 5) with angles ϕ, σ and $\pi - \phi - \sigma$. Now, σ is maximized when $\pi - \phi - \sigma = \pi/2$, from which $\sigma = \arctan\left(l/\sqrt{h^2 - l^2}\right)$, and $\phi = \arccos(l/h)$ follows.

If α is partially visible from the evader position, and the path found in Lemma 2 intersects $\partial\mathcal{R}$, then the evader should move to the last reflex vertex obstructing the path. Once reached, a new path from Lemma 2 should be computed, taking into account the time elapsed. This can be extended for the case when α is completely hidden. The evader moves in a shortest-path, until α is visible. Consider now that the evader is presented with two choices: either, to start at time t_{s1} at distance h_1 from a_α, or to start at time t_{s2} at distance h_2 from a_α, with the inequalities $t_{s1} < t_{s2}$, and $h_1 > h_2$. Given the expression for σ in Lemma 2, t_{s1} is the better choice when $l_1^2/l_2^2 > (h_1^2 - l_1^2)/(h_2^2 - l_2^2)$, for $l_i = v_e t_{si}$. Thus, given the time t_f, the best path for the evader when α is not visible can be determined.

The recontamination fans alter the computation of the paths, since a shortest-path may cross a gap visible to the pursuer. A recontamination fan is computed for every gap that merges, assuming that the region behind the original position of the gap is completely contaminated. If a path is completely contained inside the contamination regions defined by the fans, no further modification to the path is required. Otherwise, the amount of time spent crossing the region between a contamination boundary and the next merge is added to the path.

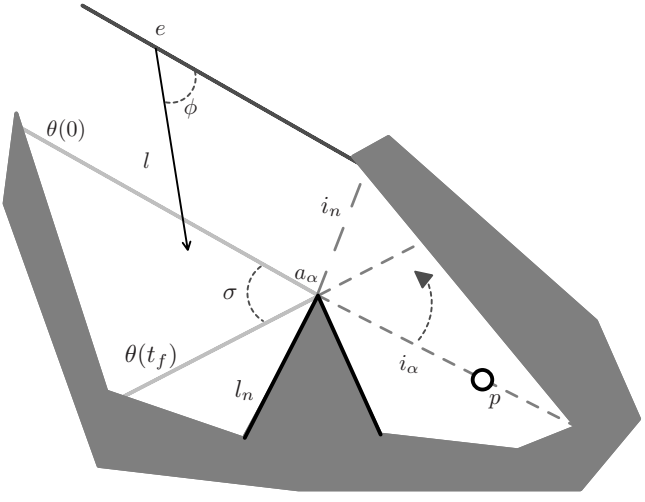

Fig. 5. Pushing a gap. The evader travels on l at angle ϕ as to maximize the angle σ, thus maximizing contamination. As the evader moves, the gap moves with it, and so does the gap extension ray i_α. In the extreme case, i_α coincides with i_n. If the pursuer crosses i_α, the evader is detected.

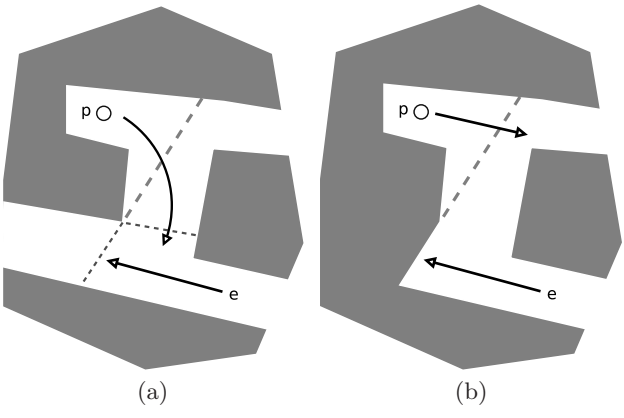

Fig. 6. Clearing equivalent an polygon. The pursuer considers the gap extension ray in (a) as the actual inflection ray. This is equivalent as clearing the polygon in (b).

Clearing an Equivalent Polygon

From the pursuer perspective, the gap extension ray i_α is a real inflection ray. When the pursuer crosses i_α, it clears an equivalent polygon in which α is one of the edges (see Figure 6). At the extreme case, α will coincide with l_n and the pursuer clears the original polygon. This holds also when the gap γ is a

nonprimitive gap. As γ moves, i_γ aligns with a bitangent complement. From the pursuer perspective, crossing i_γ before the alignment is like clearing a polygon in which γ is one of the edges, and for a pursuer strategy this polygon is equivalent to the original one. Since contamination paths are computed in a pairwise manner, this simplifies the contamination computation when the evader force a gap to split. When γ splits, say in gaps α and β, the evader could have reached α and β already, by the triangle inequality. In fact, γ will coincide with one of α or β, the one originated by the same reflex vertex. Thus the pursuer should consider, at the same time, the worst positions for α, β, and γ, since the evader may be pushing any of them.

4 The Pursuit Status

Up to now, we have described how the set $E(t)$ of contaminated regions changes as a function of time and the pursuer movements. In this section we provide an appropriate representation for the information state $\eta_t = (p(t), E(t))$. As seen in the previous section, a structure that provides the gap relations is needed. Namely, we need to know which gap will split in which other gaps. This gap hierarchy is represented with a shortest-path tree T, rooted at the pursuer position. Except the root, there is a one-to-one correspondence between the reflex vertices of $\partial\mathcal{R}$, and the nodes in T. Thus the gaps of the corresponding vertex are assigned to each node. The gap's angular position recorded at the node depends on the gap's current contamination status:

- **The gap is cleared.** If it is visible, the angle is set to the current angular position. Otherwise, it is set to the angle that aligns the gap with the bitangent complement of the merge when it was last seen.
- **The evader is pushing the gap.** The angle recorded is the one that minimizes $\theta(t)$, for the period the contamination was allowed.
- **The gap has a recontamination fan.** The angle is set to 0.

Note that even if a contaminated gap is currently visible, the angle recorded depends on the status of its contamination. The combinatorial structure of T changes as the pursuer moves. In fact, it changes in exactly the same places as gaps do [1, 16]. When T changes combinatorially, together with each merge and split the time t of the event is recorded. The value of t is necessary to compute how far the evader could have moved since the last state (i.e., in fan recontamination and pushing of gaps). The tree is modified every time a bitangent complement is crossed, an inflection ray is crossed, and when the gap extension ray of a gap being pushed is crossed. Note that if there exists a path in T between a clear gap and a contaminated one, and this path does not visit the root, then there is a contamination path between the two gaps.

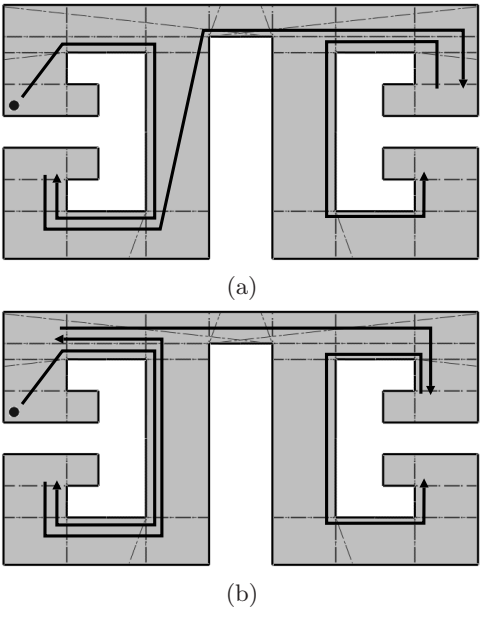

Fig. 7. Example. The path of the pursuer is shown, from the initial position marked as a black circle. (a) $v_e \ll v_p$. Note that the room at the upper-left does not get contaminated. (b) $v_e = v_p$. The upper-left room gets recontaminated, and the clearing path is longer. This example cannot be solved without bounding the speed of the evader. The algorithm presented here also finds a solution for each problem solvable without bounding the speed of the evader.

5 An Improved Pursuit Strategy

Once the representation of an information state has been defined, a search in the information space $\mathcal{I}_{ndet}$ can be performed to find a pursuit strategy. The search starting node is $\eta_0 \in \mathcal{I}_{ndet}$, which has all gaps labeled as contaminated. An information state $\eta_{t_c} \in \mathcal{I}_{ndet}$ is a search goal if it has all its gaps labeled as cleared, and no gap is being pushed. The search strategy is similar to the unbounded speed case in [4]. The visibility-cell decomposition is computed as in the unbounded speed case. The center of each cell is computed. For $\eta_t \in \mathcal{I}_{ndet}$, a set of actions U_{η_t} is defined. An action $u_\beta \in U_{\eta_t}$ takes the pursuer to a neighboring cell through a straight line. Thus, the paths are restricted to be piece-wise linear. A state is a candidate to add to the search queue when T is modified combinatorially, or when a gap extension ray is crossed, as the states are expanded with U_{η_t}. The candidate is accepted if it has at least one gap in which progress to clear it is better than in any previous state. Progress here is defined as: either the gap is cleared, or if the gap is being pushed, the angular position of the gap is bigger than in any other state.

The pursuer strategy is not complete. Nevertheless, there is an important guarantee to its performance. It is at least as powerful as any strategy for the unbounded speed case. In the unbounded speed case, there are no breaking times in the recontamination fans and the evader can move arbitrarily close to any point in the gap. When merges occur, the evader is able to transverse any shortest-path in arbitrarily small time. Finally, when it sees a gap, it can travel arbitrarily fast to the vertex that produces it. Thus, the information state correctly encodes the recontaminations for the unbounded speed case. Crossing gap extension rays becomes immediately crossing inflection rays, and the search is performed as presented originally in [4]. Figure 7 presents examples for two different speeds of the evader. These examples cannot be solved without bounding the speed of the evader.

6 Future Work

We are currently investigating the optimal strategy for the pursuer based on the description of contamination state presented in this paper. The main practical difficulty is the description of the contamination boundary of the fans. The piecewise linear approximation may be a useful tool for providing better paths for the pursuer since it is simpler to analyze. Finding the pursuers movements presents interesting challenges in optimization. For example, since the pursuer has some control in the contamination inside a fan, it can control to some extent the optimal positions for the evader once it reached a cleared gap. It may be possible to model such scenario as a zero-sum game in which the evader tries to maximize recontamination. Other interesting questions remain to be explored. For example, given that contamination travels in shortest paths, we conjecture that environments with the same shortest-path graph will require the same pursuer speed. Our future work considers the study of these questions.

References

1. Aronov, B., Guibas, L., Teichmann, M., Zhang, L.: Visibility queries in simple polygons and applications. In: Chwa, K.-Y., H. Ibarra, O. (eds.) ISAAC 1998. LNCS, vol. 1533, Springer, Heidelberg (1998)
2. Crandall, M.G., Evans, L.C., Lions, P.L.: Some properties of viscosity solutions of hamilton-jacobi equations. Trans. Amer. Math. Soc. 282 (1984)
3. Gerkey, B., Thrun, S., Gordon, G.: Clear the building: Pursuit-evasion with teams of robots. In: Proceedings of the AAAI National Conference on Artificial Intelligence (2004)
4. Guibas, L.J., Latombe, J.-C., LaValle, S.M., Lin, D., Motwani, R.: Visibility-based pursuit-evasion in a polygonal environment. In: Rau-Chaplin, A., Dehne, F., Sack, J.-R., Tamassia, R. (eds.) WADS 1997. LNCS, vol. 1272, pp. 17–30. Springer, Heidelberg (1997)
5. Guilamo, L., Tovar, B., LaValle, S.M.: Pursuit-evasion in an unknown environment using gap navigation graphs. In: IEEE/RSJ Int. Conf. on Intelligent Robots & Systems (2004)

6. Hwang, I., Stipanovic, D., Tomlin, C.J.: Polytopic approximations of reachable sets applied to linear dynamic games and a class of nonlinear systems. In: Advances in Control, Communication Networks, and Transportation Systems In Honor of Pravin Varaiya (2005)
7. Isaacs, R.: Differential Games. Wiley, New York (1965)
8. Isler, V., Daniilidis, K., Pappas, G.J., Belta, C.: Hybrid control for visibility-based pursuit-evasion games. In: IEEE/RSJ Int. Conf. on Intelligent Robots & Systems (2004)
9. Isler, V., Kannan, S., Khanna, S.: Locating and capturing an evader in a polygonal environment. In: Workshop on the Algorithmic Foundations of Robotics (2004)
10. Kameda, T., Yamashita, M., Suzuki, I.: On-line polygon search by a six-state boundary 1-searcher. Technical Report CMPT-TR 2003-07, School of Computing Science, SFU (2003)
11. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006), http://msl.cs.uiuc.edu/planning/
12. LaValle, S.M., Simov, B., Slutzki, G.: An algorithm for searching a polygonal region with a flashlight. International Journal of Computational Geometry and Applications 12(1-2), 87–113 (2002)
13. Sachs, S., Rajko, S., LaValle, S.M.: Visibility-based pursuit-evasion in an unknown planar environment. International Journal of Robotics Research (to appear, 2003)
14. Stipanovic, D.M., Hwang, U., Tomlin, C.J.: Computation of an over-approximation of the backward reachable set using subsystem level set functions. Dynamics Of Continuous Discrete And Impulsive Systems 11, 397–412 (2004)
15. Suzuki, I., Yamashita, M.: Searching for a mobile intruder in a polygonal region. SIAM J. Computing 21(5), 863–888 (1992)
16. Tovar, B., Guilamo, L., LaValle, S.M.: Gap navigation trees: Minimal representation for visibility-based tasks. In: Proc. Workshop on the Algorithmic Foundations of Robotics (2004)
17. Yavin, Y., Pachter, M.: Pursuit-Evasion Differential Games. Pergamon Press, Oxford (1987)
18. Yong, J.: On differential evasion games. SIAM J. Control & Optimization 26(1), 1–22 (1988)
19. Zaremba, L.S.: Differential games reducible to optimal control problems. In: IEEE Conf. Decision & Control, Tampa, pp. 2449–2450 (December 1989)