

Pursuit-Evasion Voronoi Diagrams in ℓ_1^*

Warren Cheung[†]

William Evans[†]

May 25, 2007

Abstract

We are given m *pursuers* and one *evader*. Each pursuer and the evader has an associated starting point in the plane, a maximum speed, and a start time. We also have a set of line segment obstacles with a total of n endpoints. Our task is to find those points in the plane, called the *evader's region*, that the evader can reach via evasive paths. A path is evasive if the evader can traverse the path from its starting point without encountering a pursuer along the way. The evader and the pursuers must obey their start time and speed constraints, and cannot go through obstacles. The partition of the plane into the evader's region and the remaining pursuers' region is called the pursuit-evasion Voronoi diagram.

We study pursuit-evasion Voronoi diagrams for the ℓ_1 metric. We show that the complexity of the diagram is $O((n+m)^2(mn+m))$ and that it can be calculated in polynomial time.

1 Introduction

We are given m pursuers and one evader. Let p_i be the start position, v_i be the velocity, and t_i be the start time of the i th pursuer. Let e , v_e , and t_e be the start position, velocity, and start time of the evader. Let O be the set of line segment obstacles that block the movement of both pursuers and evaders. Let n be the number of endpoints in O . Our task is to partition the plane into regions reachable by the evader without any chance of capture by any pursuer, and regions where capture by the pursuers is possible. This partition is called the *pursuit-evasion Voronoi diagram*.

The following definitions make this more precise. For a path ϕ and points p and q on the path ϕ . Let $d_\phi(p, q)$ be the distance along the subpath of ϕ from p to q , where $d_\phi(p, q) = \infty$ if the subpath of ϕ from p to q crosses an obstacle in O . For convenience, let $d_\phi = d_\phi(p, q)$ if ϕ is a path from p to q . Let $\tilde{d}(p, q) = \min_{\phi \in \Phi(p, q)} d_\phi(p, q)$, where $\Phi(p, q)$ is the set of all paths from p to q . In other words, $\tilde{d}(p, q)$ is the length of the shortest obstacle-avoiding path from p to q if such a path exists, or ∞ otherwise.

A path ϕ from e to p is an *evasive path* if and only if for all $q \in \phi$,

$$d_\phi(e, q)/v_e + t_e \leq \min_{i \in [m]} \tilde{d}(p_i, q)/v_i + t_i.$$

Let $d^e(p)$ be the length of the shortest evasive path to p if such a path exists, or ∞ otherwise. The evader can *reach* a point p , and p is in the evader's region, if $d^e(p) < \infty$; otherwise p is in the

*This paper is based on the first author's Master's Thesis [4]. The research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

[†]Department of Computer Science, University of British Columbia, Vancouver B.C. Canada, V6T 1Z4. Email: {wcheung, will}@cs.ubc.ca

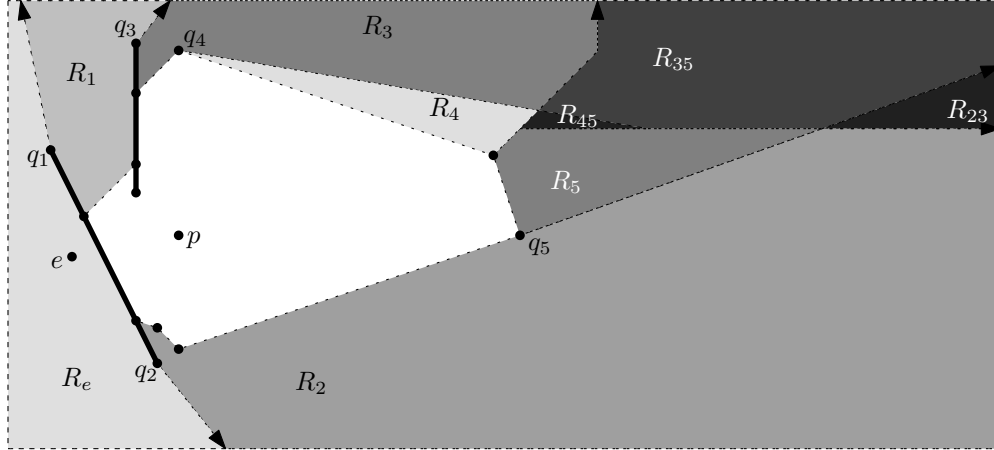


Figure 1: A simple example of the pursuit-evasion Voronoi diagram in ℓ_1 for an evader e and a pursuer p , where the evader has twice the speed of the pursuer. The thick black lines represent obstacles. Points in the region R_e can be reached by the evader taking a straight line path from e to the point in R_e . Points in R_1 can be reached by the evader taking a path from e to q_1 and then to the point in R_1 , and so on. Vertices labeled q_i are evader turning points. Regions labeled R_{ij} are equidistant from q_i and q_j . The evader region is the union of the shaded regions.

pursuers' region. See Figure 1 for an example of a pursuit-evasion Voronoi diagram using the ℓ_1 distance metric.

2 Related Work

The pursuit-evasion Voronoi diagram is a variation of the traditional Voronoi diagram that integrates properties from several Voronoi diagram types. Similar to compoundly weighted Voronoi diagrams [3, 9], each site in the pursuit-evasion Voronoi diagram has both an additive weight (start time) and a multiplicative weight (inverse of speed). Compoundly weighted Voronoi diagrams partition the plane into regions for sites, where the arrival time¹ to all points in a site's region is earliest from that site than from any other. As in the constrained Voronoi diagram [6, 1, 2], the pursuit-evasion Voronoi diagram allows obstacles (edges in the plane) and uses the shortest obstacle-avoiding path length as the distance between points.

The multiplicatively weighted crystal growth Voronoi diagram introduces the constraint that paths to a point cannot pass through points passed through earlier by another site. Schaudt and Drysdale[12, 11] describe a numerical expanding wavefront method to construct the crystal Voronoi diagram in ℓ_2 , as well as an event-driven expanding wavefront algorithm for computing the crystal Voronoi diagram for convex polygon distance functions, including ℓ_1 . Schaudt[11] shows that the multiplicatively weighted crystal Voronoi diagram has polygonal boundaries in ℓ_1 , but was unable to show a bound on the size of the diagram. He conjectured that the size of the diagram for n sites is $O(n)$. In the pursuit-evasion Voronoi diagram, the paths taken by the evader must avoid points

¹Given a speed v and a starting time t , the *arrival time* for a path of length d is $d/v + t$.

already reached by pursuers, as in the crystal Voronoi diagram, however, pursuers are allowed to move freely through the evader’s region and the pursuers’ region.

The pursuit-evasion Voronoi diagram can be viewed as the diagram obtained by growth of an agent, such as the commonly-used examples of bacteria or crystals[12, 11, 2]. However, in the case of the pursuit-evasion Voronoi diagram, we have two asymmetric types of agents: predators (i.e. pursuers), limited only by terrain and growth rate, and prey (i.e. evaders), only able to grow into empty space. The pursuit-evasion Voronoi diagram can also be applied to worst-case motion planning applications, such as camera control in telesurgery[7, 8], with the evasiveness being a guarantee against collision with uncertain pursuer motion. Another application is motion planning with both fixed obstacles and growing impediments (i.e. the pursuers), such as a spreading forest fire[10].

3 Turning Points

The points of interest in a pursuit-evasion Voronoi diagram are the points that restrict the movement of pursuers and the evader. Pursuer paths are restricted only in that they cannot cross obstacles, but evader paths cannot cross obstacles or the pursuers’ region. We define the points at which pursuer paths and evader paths might be forced to bend as turning points. We also associate a velocity and arrival time with these points since different pursuers (or the evader) might have different shortest path lengths to the points.

Definition 1. The set of *pursuer turning points* is

$$\hat{S} = \hat{P} \cup \bigcup_{i=1}^m \bigcup_{o \in O^*} \{(o, v_i, \tilde{d}(p_i, o)/v_i + t_i)\}$$

where $\hat{P} = \bigcup_{i=1}^m (p_i, v_i, t_i)$ and O^* is the set of obstacle endpoints.

Definition 2. An *evader turning point* is a triple $(p, v_e, d^e(p)/v_e + t_e)$ where p is an extreme point on the boundary of the evader’s region that is an intermediate point of a shortest evasive path, i.e., there exists a point $q \neq p$ such that a shortest evasive path ϕ from e to q has $p \in \phi$.

Notice that there is a shortest evasive path to any point in the evader’s region that bends only at evader turning points (by the triangle inequality). This path can be specified by the sequence of evader turning points it visits. Between evader turning points, the path is a straight line.

Our definition of an evader turning point requires knowledge of the evader’s region. However, we want to determine the evader’s region by calculating all evader turning points. Since the evader must avoid the pursuer, the evader is forced to “run around” regions of the plane that pursuers can reach first. These regions are obstacles to the evader and running around them increases the evasive path length. Thus, evader turning points can be obtained by calculating a succession of compoundly weighted Voronoi diagrams. We calculate the region of the plane that can be reached via a straight-line evasive path from an existing evader turning point, and then repeat this process from the evader turning points found on the boundary of this region.

Definition 3. $\text{Voronoi-Region}(\hat{p}, \hat{S})$ is the Voronoi region for the triple $\hat{p} = (p, v, t)$ in the compoundly weighted Voronoi diagram (with obstacles) for \hat{p} and the set of triples in \hat{S} .

$$\text{Voronoi-Region}(\hat{p}, \hat{S}) = \{x | \tilde{d}(p, x)/v + t \leq \min_{(p', v', t') \in \hat{S}} \{\tilde{d}(p', x)/v' + t'\}\}$$

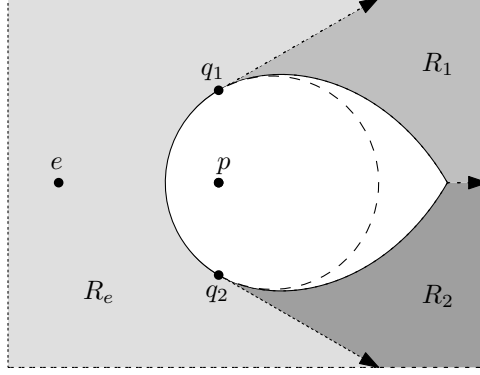


Figure 2: Pursuit-evasion Voronoi diagram in ℓ_2 for an evader e and a pursuer p , where the evader is twice as fast as the pursuer. The region R_e consists of points whose shortest evasive path is a straight line path from e . The region R_1 (resp. R_2) consists of points whose shortest evasive path contains q_1 (resp. q_2). The dashed circle is the circular boundary of the multiplicatively weighted Voronoi diagram, for comparison with the evader's boundary which is composed of two logarithmic spirals starting at q_1 and q_2 , and intersecting at the boundary between R_1 and R_2 .

If \hat{p} is an evader turning point then the evader may not have an evasive path to all points in $\text{Voronoi-Region}(\hat{p}, \hat{S})$ but it will have an evasive path to all points x such that the line segment \overline{px} lies entirely in $\text{Voronoi-Region}(\hat{p}, \hat{S})$. We call this set of points the $\text{Visible-Voronoi-Region}(\hat{p}, \hat{S})$.

We should point out that our discussion up to this point holds for any distance function that is a metric on points in the plane. However, the number of evader turning points, even for a single pursuer and evader without obstacles, can be infinite. This is the case for the Euclidean or ℓ_2 metric. A similar problem occurs in calculating the multiplicatively weighted crystal growth Voronoi diagram of Schaudt and Drysdale [12], which is closely related to the pursuit-evasion Voronoi diagram in ℓ_2 . (They are the same if there is a single pursuer who is slower than the evader and starts at the same time as the evader.)

We restrict ourselves to the study of pursuit-evasion Voronoi diagrams using the ℓ_1 metric (though our results generalize to other metrics that have polygonal unit balls). The ℓ_1 metric has the property that $\text{Visible-Voronoi-Region}(\hat{p}, \hat{S})$ is a polygon and that the evader turning points are vertices of this polygon. The $\text{Visible-Voronoi-Region}$ in ℓ_2 may involve curves such as arcs of the Apollonius circle and logarithmic spirals [12, 11] (see the example in Figure 2).

Lemma 1. *For the ℓ_1 metric, an evader turning point is the evader's starting point, an obstacle endpoint, or a vertex on the $\text{Visible-Voronoi-Region}$ of an evader turning point and the set of pursuer turning points.*

Proof. An evader turning point, $\hat{p} = (p, v_e, t)$, lies in the evader region and thus there is an evasive path ϕ from e to p that is a sequence of straight-line segments between evader turning point locations. Let $\hat{q} = (q, v_e, t')$ be the evader turning point whose location precedes p in ϕ . We claim that $\text{Visible-Voronoi-Region}(\hat{q}, \hat{S})$ has p as a vertex on its boundary. Since ϕ is an evasive path with a straight-line from q to p , p lies in $\text{Visible-Voronoi-Region}(\hat{q}, \hat{S})$ and on its boundary. Suppose p lies on the interior of a boundary edge of $\text{Visible-Voronoi-Region}(\hat{q}, \hat{S})$, then the path ϕ cannot be extended beyond p , i.e., \hat{p} is not a turning point, which is a contradiction. \square

In the next section, we bound the complexity of the pursuit-evasion Voronoi diagram in ℓ_1 by considering the number of evader turning points.

4 Size of the Pursuit-Evasion Voronoi Diagram

Lemma 2. *The number of evader turning points in a pursuit-evasion Voronoi diagram with m pursuers and n obstacle endpoints is $O((n + m)(mn + n))$.*

Proof. We start by counting the number of evader turning points (p, v_e, t) that are vertices on a Visible-Voronoi-Region between another evader turning point (p', v_e, t') and the pursuers. The turning point (p, v_e, t) is also a turning point on a Visible-Voronoi-Region between (p', v_e, t') and some pursuer turning point (o, v_i, t^*) where \overline{op} is unobstructed. (The point o is the location of the last pursuer turning point on the shortest obstacle avoiding path that the i th pursuer follows to reach p at the same time as the evader.) Since p is visible from p' , the line segment $\overline{p'p}$ is also unobstructed. Thus (p, v_e, t) is a vertex of the (compoundly weighted) bisector between (p', v_e, t') and (o, v_i, t^*) without obstacles. In ℓ_1 , vertices of this bisector occur on the x - or y -axis centred at either p' or o . If p lies on an axis centred at p' , the bisector must cross this axis at p , effectively blocking any continuation of a shortest evasive path that ends with the segment $\overline{p'p}$. Thus for p to be a turning point it cannot lie on an axis centred at p' and must lie on an axis centred at o .

If the i th pursuer is faster than the evader (i.e. $v_i > v_e$) then p is not a turning point since the faster pursuer, arriving at p at the same time as the evader, can reach any point from p before the evader. So we know that all turning points are caused by pursuers that are slower than the evader.

Let us assume for notational convenience that o is at the origin. We count all the evader turning points (p, v_e, t) that occur on the positive x -axis (the same argument applies to the other three axes). One might assume that the i th pursuer can cause at most one turning point on this axis, since the evader is faster than the pursuer and once they meet (at p) every point on the axis farther from o than p must be reachable by the faster evader before the pursuer. This ignores the possibility that the evader may be forced to deviate from the axis to “run around” a pursuer region, allowing the i th pursuer (that can run through the other pursuer’s region) to follow a shorter path and overtake the evader.

For a pursuer turning point location q and $i \in [m]$, let

$$T_i^q(x) = t_i + \tilde{d}(p_i, q)/v_i + d_{\overline{qp}}/v_i$$

be the arrival time of the i th pursuer to the point p that is on the x -axis at distance x from o when the pursuer follows a shortest obstacle-avoiding path to q and then a straight line to p . Note if \overline{qp} intersects an obstacle then $T_i^q(x)$ is infinity, if not then $d_{\overline{qp}} = |y_q - y_o| + |x_q - (x_o + x)|$ where (x_q, y_q) (resp. (x_o, y_o)) are the coordinates of q (resp. o). Figure 3 shows an example of a function $T_i^q(x)$. The function is linear with slope $-1/v_i$ for $x < x_q$ and is linear with slope $+1/v_i$ for $x > x_q$, except for those values of x where it is infinite.

Let $T_{\min}^q(x) = \min_{i \in [m]} T_i^q(x)$. We are interested in evader turning points, where the evader reaches a point p on the x -axis at the same time as the earliest pursuer reaches that point via o . The function $T_{\min}^o(x)$ is the earliest pursuer arrival time on the x -axis when the pursuers must arrive via o . Since $T_i^o(x)$ is linear with slope $+1/v_i$ (since $x_o = 0$), $T_{\min}^o(x)$ is piecewise linear with positive, non-increasing slope. (See Figure 4 for an example).

Let $T_{\min}(x) = \min_q T_{\min}^q(x)$ be the earliest pursuer arrival time on the x -axis when the pursuers may arrive via any pursuer turning point. The function $T_{\min}(x)$ is continuous and piecewise linear

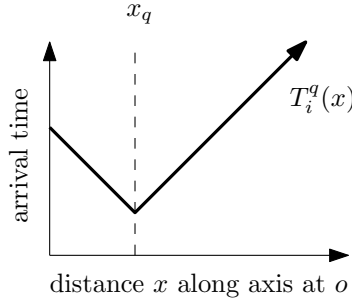


Figure 3: $T_i^q(x)$ is the arrival time of the i th pursuer to the point at distance x from o on the positive x -axis (with origin o), when the pursuer takes a path via the point $q = (x_q, y_q)$. The slope is $\pm 1/v_i$.

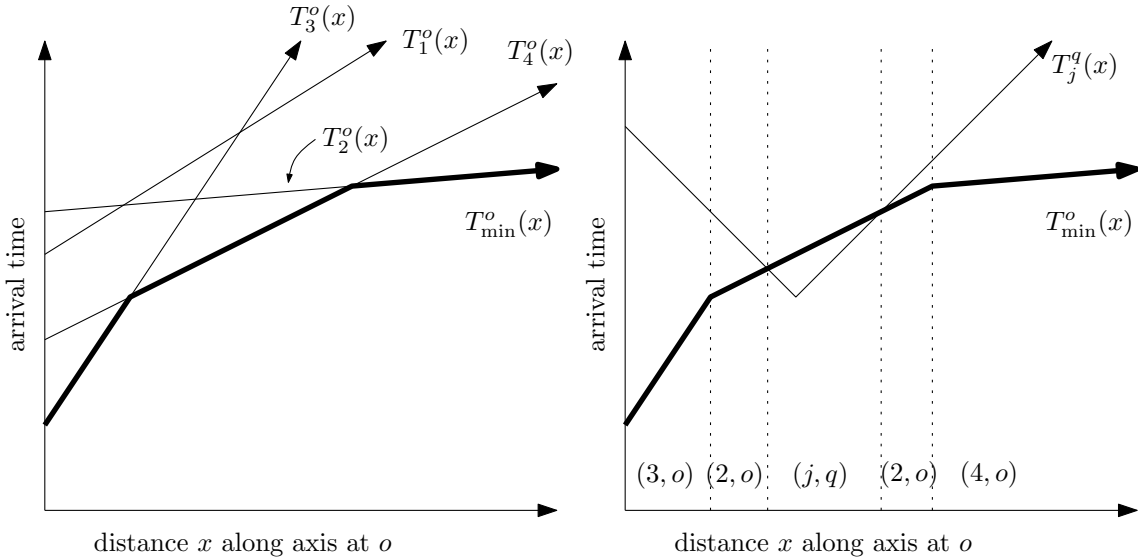


Figure 4: The thick black line is $T_{\min}^o(x)$, the minimum arrival time of any pursuer to the positive x -axis originating at o , when the pursuers take paths via o . $T_i^o(x)$ is the arrival time for the i th pursuer. On the right, at most one interval labeled (j, q) (for $j \in [m]$ and $q \in \hat{S}$) can share its right endpoint with an interval labeled (i, o) for $i \in [m]$.

and we use it to partition the x -axis into intervals. We label with (i, q) every contiguous interval of the x -axis where $T_{\min}(x) = T_i^q(x)$. We may label several disjoint intervals with the same label since $T_i^q(x)$ may achieve the minimum at different intervals of the x -axis.

We only count evader turning points that occur within intervals labeled (i, o) for $i \in [m]$. If an interval (i, q) with $q \neq o$ contains an evader turning point it will be counted when we consider the axis originating at q . Each interval labeled (i, o) contains at most one evader turning point since if p is the location of an evader turning point in that interval, all points p' farther than p from o will be reached by the evader before pursuer i . (Recall that pursuer i must be slower than the evader in order to create an evader turning point, and, in the interval labeled (i, o) , no other pursuer can create a region that the evader must “run around”.)

We count the number of (i, o) intervals by counting the number of values x that can be the left endpoint (or lower bound) of such an interval. If x^* is the left endpoint of an (i, o) interval, it is the right endpoint of a (j, q) interval. $T_j^q(x)$ has increasing slope at x^* since x^* is the right endpoint of the (j, q) interval and $T_j^q(x)$ intersects $T_{\min}^o(x)$ at x^* . Since $T_{\min}^o(x)$ has non-increasing slope, $T_j^q(x)$ cannot equal $T_{\min}^o(x)$ for any $x > x^*$ (since it is greater than $T_{\min}^o(x)$ for all $x > x^*$). Thus (j, q) can be adjacent (on the left) to at most one (i, o) interval. Thus the number of (i, o) intervals on the positive x -axis (with origin o) is at most $mn + m$ (the number of pairs (j, q)). Since there are $m + n$ pursuer turning point locations, there are $m + n$ possible origins o and at most $4(m + n)(mn + m)$ such evader turning points (counting the four axes).

By Lemma 1, there are at most $n + 1$ other evader turning points that are not vertices on a Visible-Voronoi-Region between another evader turning point and the pursuers. The lemma follows. \square

Note that if all the pursuers have the same speed, the number of evader turning points is $O(n + m)$.

Theorem 1. *The pursuit-evasion Voronoi diagram has at most $O((n + m)^2(mn + m))$ vertices where m is the number of pursuers and n is the number of obstacle endpoints.*

Proof. Lemma 2 counts evader turning points. It remains to count vertices that are on the boundary of the evader’s region that are not turning points. These vertices may be obstacle endpoints (which have already been counted as potential evader turning points), meeting points on the axes centred at an evader turning point, or meeting points with a faster pursuer on the axes centred at a pursuer turning point. Here a *meeting point* is a point on the boundary of the evader’s region that is reached at the same time by the evader and a pursuer.

A meeting point on an axis centred at an evader turning point arises from a pursuer intercepting a shortest evasive path that follows the axis from the evader turning point. This is a point of capture that blocks the evader from following the axis any further. Thus there are no other meeting points along this axis centred at this evader turning point. Since each evader turning point has four axes, the number of these meeting points is at most four times the number of evader turning points, which is $O((n + m)(mn + m))$.

A meeting point between the evader and a faster pursuer is not an evader turning point since once a faster pursuer meets the evader, the evader cannot follow any path from that point that is evasive. We perform the following conservative analysis to upper-bound the number of such interactions.

For each pursuer turning point location o , we consider how many times a pursuer taking a path via o can meet an evader along an axis centred at o . The evader must come in a straight line from

some evader turning point, q , to such a meeting point. Recall from the proof of Lemma 2 that $T_{\min}^o(x)$ is the arrival time on the positive x -axis of the earliest arriving pursuer via o , and that it has positive, non-increasing slope. The arrival time of the evader from q is a function

$$T^q(x) = t_e + d^e(q)/v_e + d_{\overline{qp}}/v_e$$

where p is the point at distance x from o on the positive x -axis. As with the functions $T_i^q(x)$, this function is linear with slope $-1/v_e$ for $x < x_q$ and slope $+1/v_e$ for $x > x_q$. Hence, it intersects $T_{\min}^o(x)$ at most twice and may cause at most two meeting points. There are $m+n$ pursuer turning point locations and at most $4(m+n)(mn+m)$ evader turning points, so the number of such meeting points is at most $32(n+m)(mn+m)(n+m) = O((n+m)^2(mn+m))$. As a result, the evader's region has $O((n+m)^2(mn+m))$ vertices. \square

5 Algorithm

We describe how the pursuit-evasion Voronoi diagram can be computed via construction of a shortest path tree.

Definition 4. A *shortest path tree* for the evader is a tree whose root is (e, v_e, t_e) and whose nodes are evader turning points. A node (p', v_e, t') is the parent of (p, v_e, t) if a shortest evasive path from e to p contains p' . (If a node has multiple parents according to this definition, any one may be chosen.)

Thus, if q is in the evader's region there is a shortest evasive path (e, q_1, \dots, q_k, q) where $((e, v_e, t_e), (q_1, v_e, d^e(q_1)/v_e + t_e), \dots, (q_k, v_e, d^e(q_k)/v_e + t_e))$ is a path in the shortest path tree.

To construct the shortest path tree, we maintain an incomplete shortest path tree T and a *fringe* F of points that can be reached via straight-line paths from the leaves of T and are possible evader turning point locations. We iteratively add the "closest" fringe point, p , to T and update the fringe to include those possible evader turning point locations reachable from p via a straight line. We repeat this process until no points remain in F . As each new tree node (p, v_e, t) is added to T , we associate with it the evader region consisting of all evasive paths that end with a straight-line segment from p , and add this region to the evader region, \mathcal{E} . Once the shortest path tree is complete, we will have obtained all regions reachable via straight-line segments from evader turning points, identifying the entirety of the evader region.

Let F denote the set of potential evader turning point locations that are reachable via an evasive shortest path that ends with a straight-line segment from a leaf node in T . With each $p \in F$, we keep an earliest estimated arrival time $t[p]$ and parent $\pi[p]$. Initially, $F := \{e\}$, $t[e] := t_e$, $\pi[e] := \emptyset$, $T := \emptyset$, and the evader region $\mathcal{E} := \emptyset$.

The shortest path condition for the nodes in the shortest path tree is ensured by the application of Dijkstra's algorithm. The node dequeued and added to the tree is the node with the shortest estimated arrival time (lines 2,3).

For each new tree node \hat{p} (added to T in line 3), it is necessary to remove all points in F that are no longer possible evader turning point locations (line 4), i.e., all points q in F where $t[q] > t[\hat{p}] + d_{\overline{pq}}/v_e$. Some of these points, in particular obstacle endpoints, may be added to F again (in line 5), but with \hat{p} as parent.

Once a new evader turning point \hat{p} is added to T , the visible Voronoi region R for \hat{p} is computed. Let B be the set of vertices forming the boundary of R . We add to F those vertices of B that are

Algorithm 1 Outline of pursuit-evasion Voronoi diagram computation

- 1: **while** F not empty **do**
 - 2: Dequeue evader turning point location p from F where for all $q \in F$, $t[p] \leq t[q]$.
 - 3: Add $\hat{p} = (p, v_e, t[p])$ to T with parent $\pi[p]$.
 - 4: Remove from F any point $q \in F$ where $t[q] > t[p] + d_{\overline{pq}}/v_e$
 - 5: Add each potential evader turning point location x from $R := \text{Visible-Voronoi-Region}(\hat{p}, \hat{S})$ to F with parent $\pi[x] := p$ and $t[x] := t[p] + d_{\overline{px}}/v_e$.
 - 6: $\mathcal{E} := \mathcal{E} \cup R$
 - 7: **end while**
-

potential evader turning point locations and whose arrival time via a straight line from \hat{p} is less than the arrival time via a straight line from \hat{q} for all \hat{q} in T . A vertex x of B is a potential evader turning point if there exists a point $y \neq x$ such that $x \in \overline{py} \in R$. This can be determined during the calculation of R .

6 Time Complexity of the Algorithm

Lemma 3. *For any evader turning point \hat{p} , $\text{Visible-Voronoi-Region}(\hat{p}, \hat{S})$ has $O(mn^2)$ vertices and can be calculated in time $O(mn^2 \log(mn))$.*

Proof. Let $\text{Visible}(p, O)$ be the region visible from point p with respect to obstacles O . Let $\text{Bisector-Region}(\hat{p}, \hat{q})$ be the Voronoi region for \hat{p} in the (compoundly weighted) Voronoi diagram of the two triples \hat{p} and \hat{q} ignoring obstacles. $R := \text{Visible-Voronoi-Region}(\hat{p}, \hat{S})$ can be computed as follows:

$$C := \bigcup_{\hat{q} \in \hat{S}} \{\text{Bisector-Region}(\hat{q}, \hat{p}) \cap \text{Visible}(q, O)\}$$
$$R := \text{Visible}(p, C \cup O)$$

First, the Bisector-Region for each of the $mn + m$ pursuer turning points in \hat{S} against the evader turning point \hat{p} is computed. Each of these regions has size at most 6 vertices, the maximum size of a two point Voronoi diagram in ℓ_1 , and can be calculated in constant time. Obstacles are ignored as the intersection with $\text{Visible}(q, O)$ will take care of them. Pursuer turning point visibility is only limited by obstacles, therefore, each $\text{Bisector-Region}(\hat{q}, \hat{p}) \cap \text{Visible}(q, O)$ has $O(n)$ vertices and can be computed in time $O(n \log n)$ [13]. Thus C has a total of $O(n(mn + n))$ vertices.

The evader turning point visibility treats O and C as obstacles since both can block the evader. We treat $C \cup O$ as a set of $O(n(mn + n))$ line segments when calculating the visible region from the evader turning point. Thus calculating R takes $O(n(mn + n) \log(n(mn + n)))$ time and R has complexity $O(mn^2)$. \square

Note that this method of computing Visible-Voronoi-Region allows computation of the pursuit-evasion Voronoi diagram not only for ℓ_1 , but for any distance metric where Bisector-Region results in a polygonal region (though the complexity may be affected).

Theorem 2. *The running time of the algorithm is at most $O((n + m)^2 n^4 m^3 \log(mn))$.*

Proof. Let $\alpha(n, m)$ be the cost of the computing the Visible-Voronoi-Region for an evader turning point and m pursuers among obstacles with a total of n endpoints, and let $\tau(n, m)$ denote the number of evader turning points.

The algorithm has $\tau(n, m)$ iterations, with each node of the shortest path tree being dequeued from F exactly once.

During each iteration, the region $R := \text{Visible-Voronoi-Region}(\hat{p}, \hat{S})$ is computed in time $\alpha(n, m)$. According to Lemma 3, R has size $O(mn^2)$. In line 5, it is necessary to compute the straight-line distance between each vertex in R and every evader turning point currently in the shortest path tree, which takes $O(\tau(n, m)mn^2)$ time. In line 4, we compute the straight-line distance between the $O(\tau(n, m)mn^2)$ potential turning point locations in F and the new evader turning point \hat{p} . Each distance calculation checks for obstacle intersection and, hence, takes $O(\log n)$ time. In line 6, the region \mathcal{E} has complexity $O((n + m)^2(mn + m))$ (by Theorem 1) and thus the update takes $O((n + m)^2(mn + m) \log(mn))$ [5]. Therefore, the cost for each new evader turning point is at most $\alpha(n, m) + O(\tau(n, m)mn^2 \log(mn))$ for a total cost of $\tau(n, m)(\alpha(n, m) + O(\tau(n, m)mn^2 \log(mn)))$. By Lemmas 2 and 3, this total cost is no more than $O((n + m)^2 n^4 m^3 \log(mn))$. \square

7 Conclusion

We show that, when using the ℓ_1 distance metric, the size of the pursuit-evasion Voronoi diagram for an evader is polynomial in the number of pursuers and obstacles, and we give a polynomial time algorithm to compute this diagram. We use a variation of Dijkstra’s algorithm for computing single source shortest paths to explore the plane, while computing regions reachable by the evader. This allows us to consider the pursuers and the evader, each with their own starting point, starting time, and speed, as well as line segment obstacles in the plane.

The pursuit-evasion Voronoi diagram describes the region for a single evader in the presence of multiple pursuers. Given multiple evaders, the region reachable by all of the evaders is simply the intersection of the regions reachable by each of the individual evaders separately.

Although the algorithm does not directly generalise to ℓ_2 — the curves in the ℓ_2 pursuit-evasion Voronoi diagram would require the equivalent of an infinite number of evader turning points — we can bound the ℓ_2 diagram using the ℓ_1 pursuit-evasion Voronoi diagram. Since $\frac{1}{\sqrt{2}}\ell_1$ distance $\leq \ell_2$ distance $\leq \ell_1$ distance, computing the ℓ_1 pursuit-evasion Voronoi diagram with the evader’s speed multiplied by $\sqrt{2}$ gives an evader region that contains the ℓ_2 evader region. Similarly, multiplying the pursuers’ speeds by $\sqrt{2}$ gives an ℓ_1 evader region that is contained in the ℓ_2 evader region.

References

- [1] Takao Asano and Tetsuo Asano. Voronoi diagram for points in a simple polygon. In *Proceedings of the Japan-US Joint Seminar, 1986, Discrete Algorithms and Complexity*, pages 51–64, 1987.
- [2] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, September 1991.
- [3] F. Aurenhammer and H. Edelsbrunner. An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recognition*, 17:251–257, 1984.

- [4] Warren Cheung. Constrained pursuit-evasion problems in the plane. Master's thesis, University of British Columbia, September 2005.
- [5] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational geometry: algorithms and applications*, pages 33–40. Springer-Verlag New York, Inc., Secaucus, NJ, USA, second, revised edition, 2000.
- [6] W. Randolph Franklin, Varol Akman, and Colin Verrilli. Voronoi diagrams with barriers and on polyhedra for minimal path planning. *The Visual Computer*, 1(2):133–150, 1985.
- [7] Kei Kobayashi and Kokichi Sugihara. Crystal Voronoi diagram and its applications to collision-free paths. In *ICCS '01: Proceedings of the International Conference on Computational Sciences-Part I*, pages 738–747, London, UK, 2001. Springer-Verlag.
- [8] T. Li and T. Yu. Planning tracking motions for an intelligent virtual camera. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1353–1358, 1999.
- [9] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial tessellations: Concepts and applications of Voronoi diagrams*, pages 128–138. Wiley & Sons, 1992.
- [10] Waldir L. Roque and Howie Choset. The green island formation in forest fire modeling with Voronoi diagrams. Abstract for *3rd CGC Workshop on Computational Geometry*, 1998.
- [11] Barry Schaudt. *Multiplicatively Weighted Crystal Growth Voronoi Diagrams*. PhD thesis, Dartmouth College, 1992.
- [12] Barry Schaudt and Robert L. (Scot) Drysdale III. Multiplicatively weighted crystal growth Voronoi diagrams (extended abstract). In *Symposium on Computational Geometry*, pages 214–223, 1991.
- [13] Subhash Suri and Joseph O'Rourke. Worst-case optimal algorithms for constructing visibility polygons with holes. In *SCG '86: Proceedings of the second annual symposium on Computational geometry*, pages 14–23, New York, NY, USA, 1986. ACM Press.