

Environment Characterization for Non-Recontaminating Frontier-Based Robotic Exploration

Mikhail Volkov, Alejandro Cornejo, Nancy Lynch, and Daniela Rus

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge, MA 02139
{mikhail, acornejo, lynch, rus}@csail.mit.edu

Abstract. This paper addresses the problem of obtaining a concise description of a physical environment for robotic exploration. We aim to determine the number of robots required to clear an environment using non-recontaminating exploration. We introduce the medial axis as a configuration space and derive a mathematical representation of a continuous environment that captures its underlying topology and geometry. We show that this representation provides a concise description of arbitrary environments, and that reasoning about points in this representation is equivalent to reasoning about robots in physical space. We leverage this to derive a lower bound on the number of required pursuers. We provide a transformation from this continuous representation into a symbolic representation. Finally, we present a generalized pursuit-evasion algorithm. Given an environment we can compute how many pursuers we need, and generate an optimal pursuit strategy that will guarantee the evaders are detected with the minimum number of pursuers.

Keywords: swarm robotics, frontier-based exploration, distributed pursuit-evasion, environment characterization, mathematical morphology, planar geometry, graph combinatorics, game determinacy

1 Introduction

This paper deals with the problem of developing a concise representation of a physical environment for robotic exploration. We address a specific type of exploration scenario called pursuit-evasion. In this scenario, a group of robots are required to sweep an unexplored environment and detect any intruders that are present. Pursuit-evasion is an example of non-recontaminating exploration, whereby an initially unexplored and contaminated region is cleared while ensuring that the cleared region does not become contaminated again. Pursuit-evasion is a useful model for many applications such as surveillance, security and military operations. Aside from the classical pursuit-evasion problem, there are other scenarios that motivate non-recontaminating exploration. One example is cleaning up an oil spill in the ocean using robots, where the oil can leak into previously decontaminated water whenever part of the frontier is not guarded by a robot. Another example is a disaster response scenario such as after an earthquake, where a group of robots is required to locate all survivors in some inaccessible area, while the survivors are possibly moving in the environment.

1.1 Related Work

Visibility-based pursuit-evasion in continuous two-dimensional space was first introduced in [25]. Frontier-based exploration was introduced in [28] and extended to multiple robots in [29]. In [7] the authors consider limited visibility frontier-based pursuit-evasion in non-polygonal environments, making use of the fact that not allowing recontamination means we do not need to store a map of the environment; here a distributed algorithm is presented which works by locally updating the frontier formed by the sensor footprints of the robots. Another distributed model was more recently considered in [3]. Limited visibility was also considered in [23] which presents an algorithm for clearing an unknown environment without localization. In [8] the problem was considered for a single searcher in a known environment. Generally speaking, these algorithms do the correct thing locally, but do not rely on, or make any guarantees for, a global description of an environment. As such, they may not be guaranteed to terminate.

Environment characterization for pursuit-evasion has been considered for polygonal spaces. In [11], [12] pursuit-evasion in connected polygonal spaces is investigated, and tight bounds derived on the number of pursuers necessary based on the number of polygon edges and holes. In [19] basic environment characterization is established, by means of a general decomposition concept based on a finite complex of conservative cells. In [27] similar ideas were explored, and several metrics proposed for primitive characterization of polygonal spaces. We highlight that these studies considered robots equipped with infinite range visibility sensors deployed in polygonal spaces. Consequently the scope of environment characterization was limited. By contrast we consider limited visibility sensors, and we do not require the environment to be polygonal.

The medial axis has previously been studied in the context of robotic navigation. For example, in [10], [13], [26] the medial axis was used as a heuristic for probabilistic roadmap planning. One of the key features that makes the medial axis attractive for such applications is that it captures the connectivity of the free space. In this work, we use the medial axis to similarly capture the underlying topological and geometric properties of our environment, and use it to transform the problem into a graph formulation.

Pursuit-evasion on graphs that are representations of some environment goes back as early as [20], [21]. Randomized pursuit strategies on a graph are considered in [1]. Roadmap-based pursuit-evasion is considered in [14] and [22] where the pursuer and evader share a map and act according to different rules. In [22] a graph-based representation of the environment is used to derive heuristic policies in various scenarios. More recently, [17] presents a graph-based approach to the pursuit-evasion problem whereby agents use blocking or sweeping actions to detect all intruders in the environment. In [15] and [16] the more general graph variant of the problem was reduced to a tree by blocking edges. Although discrete graph-based models offer termination and correctness guarantees, they assume the world is suitably characterized and make no reference to the underlying physical geometry of the environment that is being represented.

The aim of this paper is to bridge the different levels of abstraction that work to date has been grounded in. We make use of the medial axis as a configuration space to derive a robust representation of a continuous environment that captures its underlying properties. We use the medial axis leveraged by existing exploration models to build a concise representation of arbitrary environments in continuous two-dimensional space.

We then transform this representation into the discrete domain. First, this allows us to calculate bounds on the number of pursuers required to clear an environment, and to provide termination guarantees for existing pursuit-evasion algorithms. Second, this establishes an application platform for existing graph-based algorithms making them applicable to continuous environment descriptions.

1.2 Paper Organization

This paper is organized as follows. In Section 2 we provide a formal model for non-recontaminating exploration and state the problem that we are addressing. In Section 3 we introduce the medial axis as a configuration space and show that reasoning about points in this space is equivalent to reasoning about robots in the physical world. We formalize the notion of width, corridors and junctions and derive bounds on the number of robots required to traverse a junction. In Section 4 we present a transformation from this continuous configuration space into a symbolic representation in the discrete domain. Finally, we present an optimal pursuit-evasion algorithm, and prove correctness for this algorithm.

2 Problem Formulation

We now present a formal model of the problem we are addressing. Our model builds on the notation and terminology introduced in [7]. We have a team of n exploration robots deployed in the Euclidean plane \mathbb{R}^2 . Each robot is equipped with a holonomic (uniform in all orientations) sensor that records a line of sight perception of the environment within a maximum sensing radius r . We assume that two robots can reliably communicate if they are within line of sight of each other and if the distance between their positions is less than or equal to $2r$.

The position of a robot is constrained to be within some *free region* Q , which is a closed compact subset of \mathbb{R}^2 . The *obstacle region* B makes up the rest of the world, and is defined as the complement of Q . In this paper we require both Q and B to be connected spaces, which means there are no holes in the environment. We define the *obstacle boundary* ∂B as the oriented boundary of the obstacle region (which by definition is the oriented boundary of the free region).

We assume a continuous time model, i.e. time $t \in \mathbb{R}_{\geq 0}$. Let H_t^i be the holonomic *sensor footprint* of robot i at time t , which is defined as the subset of Q that is within direct line of sight of robot i and within distance r of robot i . Formally, if $p \in \mathbb{R}^2$ is the position of robot i at time t , then $H_t^i = \{x \in Q \mid d(p, x) \leq r \wedge \forall y \in [px], y \in Q\}$ where $d(x, y)$ is the Euclidean distance between x and y (see Fig. 1a). Let H_t be the union of the sensor footprints of all robots at some time t , given by $H_t = \bigcup_{i=0}^n H_t^i$. This corresponds to the region being sensed by the robots at time t . We define the *inspected region* $I_t \subseteq Q$ as the union at time t of all previously recorded sensor footprints, given by $I_t = \{p \in \mathbb{R}^2 \mid \exists t_0 \in [0, t] \text{ such that } p \in H_{t_0}\}$. The *contaminated region* (or unexplored region) U_t is defined as the free space that has not been inspected by time t , given by $U_t = Q \setminus I_t$ (see Fig. 1b). Note that at time $t = 0$ the contaminated region is given by $Q \setminus H_0$. We define the *cleared region* $C_t \subseteq I_t$ as the inspected region that is not currently being sensed, given by $C_t = I_t \setminus H_t$. We say that *recontamination* occurs at time t if the cleared region C_t comes in contact with the contaminated region U_t (we

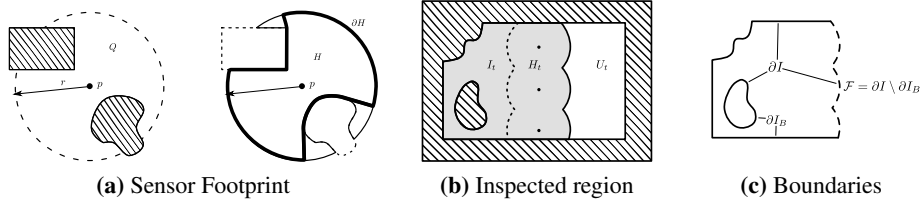


Fig. 1: Fig. 1a shows a robot located at some position p within the free region Q . The sensor footprint H and its oriented boundary ∂H are shown on the right. Fig. 1b shows the inspected region I_t and the contaminated region U_t at some time t . Fig. 1c shows the inspected region boundary ∂I , the inspected obstacle boundary ∂I_B and the frontier \mathcal{F} .

understand two regions to be in contact if the intersection of their closure is non-empty, i.e. $\text{cl}(C_t) \cap \text{cl}(U_t) \neq \emptyset$.

When time is clear from context, we understand I and U to mean the current inspected region and the current contaminated region, respectively. We define the *inspected region boundary* ∂I as the oriented boundary of the inspected region I . We define the *inspected obstacle boundary* ∂I_B as the intersection of the inspected region boundary and the obstacle boundary, given by $\partial I_B = \partial I \cap \partial B$. We define the *frontier* \mathcal{F} as the free (non-obstacle) boundary of the inspected region, given by $\mathcal{F} = \partial I \setminus \partial I_B$ (see Fig. 1c). Observe that by definition the frontier \mathcal{F} separates the free region Q into the inspected region I and the contaminated region U . Observe also that the frontier need not be connected, and is in general the union of one or more disjoint maximally connected arcs. We understand the frontier of a group of robots $\mathcal{F}' \subseteq \mathcal{F}$ to mean a single maximally connected arc of the total frontier formed by the exterior boundary of the sensor footprints of that group of robots.

The goal of exploration algorithms is to inspect the entire free region. For non-recontaminating exploration the goal is to inspect the entire free region without admitting recontamination. In both cases we say that an environment has been successfully explored if $I_t = Q$ at some time t . In this paper we deal specifically with non-recontaminating exploration and present an algorithm that is guaranteed to explore a space without admitting recontamination.

An algorithm for exploration relies on robots to “expand” the frontier boundary until the entire free region becomes inspected. However, in a non-recontaminating exploration algorithm the goal is not only to expand, but also to “guard” the frontier, ensuring that the inspected region does not become contaminated again. This difference makes non-recontaminating exploration more restrictive than conventional exploration. For example, observe that regardless of the size of the sensing radius of the robots (as long as $r > 0$), or the properties of the world (as long as Q is a connected space), a single robot can always explore the world. However, in non-recontaminating exploration this is not true in general. Informally speaking, if the “width” of the corridors in the free region is larger than the sensing radius of a robot, then it should appear obvious that a single robot cannot simultaneously expand and guard the frontier to inspect the entire free region.

Consider the simple rectangular free region Q shown in Fig. 1b. We can reason that if the width of Q is less than the sum of the sensor diameters of the n robots, then the environment can be explored without admitting recontamination. However, even in this simple example it is not completely clear what is meant by width. Notice that if we consider width to be the distance from the left to the right border then this reasoning fails — in this case width would specifically mean the smaller of the two dimensions. So it is already non-trivial how to characterize a very simple environment, and things become much more complicated in non-rectangular environments.

In this paper we study the relationship between an environment Q , the sensor radius r , and the number of robots n required for non-recontaminating exploration of Q . Intuition tells us that corridor width and junctions are important features. We formalize the notion of corridors and junctions and present a general method for computing a configuration space representation of the environment that captures this intuition. We show that this representation provides a concise description of arbitrary environments.

A canonical example of non-recontaminating exploration is pursuit-evasion. In this scenario there is a group of robot pursuers and a group of robot evaders deployed in the free region Q . The evaders are assumed to be arbitrarily small and fast. The goal of the pursuers is to catch the evaders (by detecting their presence within the sensor footprint), and the goal of the evaders is to avoid getting caught. Whenever part of the frontier is not being guarded by the pursuers, the evaders can move undetected from the contaminated region to the previously inspected region, thereby recontaminating it.

3 Environment Analysis

In this section we present the medial axis as a configuration space and show that reasoning about points in this configuration space is equivalent to reasoning about robots in physical space. First, we establish the necessary geometric framework, accompanied by a series of definitions and claims. Second, we introduce an exploration model in this configuration space and justify that it allows us to reason about the physical movement of the robots in the environment.

3.1 Environment Geometry

The *distance transform* is a mapping $\mathcal{D} : \mathbb{R}^2 \rightarrow \mathbb{R}$ where $\mathcal{D}(x) = \min_{y \in B} \{d(x, y)\}$ and $d(x, y)$ is the Euclidean distance between x and y (extending definition in [5] to the continuous domain) (see Fig. 2b). Observe that by definition if $x \notin Q$ then $\mathcal{D}(x) = 0$. The distance transform of a point $x \in Q$ captures the notion of “undirected width” of a region around a point x in free space, that is we get a measure of how wide or narrow a region is without being explicit about orientation.

The *medial axis* or *skeleton* S of a free space is defined as the locus of the centers of all maximal inscribed circles in the free space [4] (see Fig. 2c). Equivalently, the skeleton can be defined as the locus of quench points of a fire that has been set to a grass meadow at all points along its boundary [2], [24]. The skeleton captures the topology of the free space, and aids us in determining which parts of an environment should be considered “corridors” and which parts should be considered “junctions” of multiple corridors.

The *degree* of a point $x \in S$ is given by the function $\theta : S \rightarrow \mathbb{N}_{>0}$ which maps every point on the skeleton to a natural number k . Specifically, we define a point $x \in S$

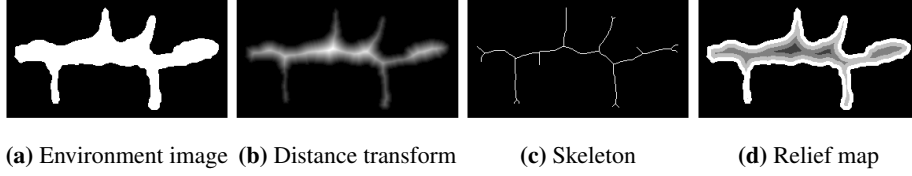


Fig. 2: The environment is represented by a binary image in Fig. 2a. Fig. 2b shows the distance transform \mathcal{D} of the environment. Fig. 2c shows the skeleton S . Fig. 2d shows the relief map quantization. The relief contours indicate multiples of the sensing radius r .

to have degree $\theta(x) = k$ if there exists an $a \in \mathbb{R}_{>0}$ such that $\forall \varepsilon \in (0, a]$ a circle centered at x of radius ε intersects the skeleton S at exactly k points.

Borrowing notation from [6], we use the degree of a point $x \in S$ to distinguish between three types of points on the skeleton: corridor points, end points and junction points. Specifically, for a point $x \in S$, we say x is an *end point* if $\theta(x) = 1$, x is a *corridor point* if $\theta(x) = 2$, and x is a *junction point* if $\theta(x) > 2$ (see Fig. 3a). We refer to a continuous arc of corridor points on the skeleton simply as a *corridor*.

An alternative definition for $\theta(\cdot)$ can be stated as follows. For a point $x \in S$ let C be the maximal inscribed circle centered at x , and let G be the intersection of this circle with the obstacle boundary, given by $G = C \cap \partial B$. (Observe that by definition C has radius $\mathcal{D}(x) \neq 0$, and since C is maximal, G is non-empty.) Then $\theta(x)$ is defined as the number of maximally connected arcs in G . This definition for $\theta(\cdot)$ is equivalent to the previous one [4], [9], [18].

Let $G_1, G_2, \dots, G_{\theta(x)}$ be the set of maximally connected arcs of G . Note that in most cases these arcs are in fact just single points, which corresponds to the intuitive notion of the circle being tangent to the boundary at these points. A cursory glance reveals that this is the case for most corridor points and junction points. For end points that lie on the obstacle boundary, the tangent point coincides with the end point itself. However, the generality is necessary in a few special cases, such as end points of regions that taper off in a sector. In these cases the maximal inscribed circle C will be tangent to the obstacle boundary at a continuous arc segment of points. In order to simplify the discussion we define the tangent points $\tau_1(x), \tau_2(x), \dots, \tau_{\theta(x)}(x)$ of a point $x \in S$ as the midpoints of the tangent arcs $G_1, G_2, \dots, G_{\theta(x)}$ (see Fig. 3b).

We define a *boundary wall* as a maximally connected arc segment of the obstacle boundary ∂B that does not contain a tangent point of any end point. Formally $\partial B_0 \subset \partial B$ is a boundary wall if it is a maximally connected arc segment such that $\forall e \in S \mid \theta(e) = 1, \tau(e) \notin \partial B_0$.

Lemma 1. *For a junction point $j \in S$, the $\theta(j)$ tangent points of j are located on $\theta(j)$ distinct boundary walls.*

Proof. A circle C of radius $\mathcal{D}(j)$ centered at a junction point j will be tangent to the obstacle boundary at the $\theta(j)$ tangent points of j . Each pair of adjacent tangent points $\tau_i, \tau_{i+1} \in C$ (in the sense of counter-clockwise orientation along C) will be on opposite sides of one corridor. Consider the obstacle boundary arc segment $[\tau_i \tau_{i+1}]$ (in the sense

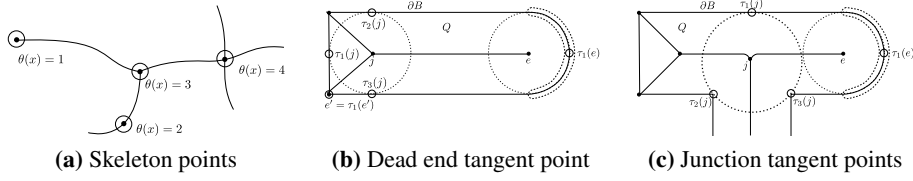


Fig. 3: Fig. 3a shows the three types of points on the skeleton: end points, corridor points, and junction points. A corridor is a continuous arc of corridor points. Fig. 3b shows three points on the skeleton e , e' , j and their respective tangent points. For end point e the tangent point $\tau_1(e)$ is the midpoint of the tangent arc segment shown (dotted outline). For end point e' the tangent point $\tau_1(e')$ coincides with the end point itself. For junction point j there are $\theta(j) = 3$ tangent points τ_1, τ_2, τ_3 . Fig. 3c shows a junction point j of degree $\theta(j) = 3$. All 3 tangent points τ_1, τ_2, τ_3 are located on distinct boundary walls.

of counter-clockwise orientation along ∂B). If e is the end point of the corridor that straddles the interior of $[\tau_i \tau_{i+1}]$, then $\tau(e)$ will lie on $[\tau_i \tau_{i+1}]$. Thus the boundary wall containing τ_i is disjoint from the boundary wall containing τ_{i+1} since neither contains $\tau(e)$. Since this applies for each pair of adjacent tangent points, we conclude that all $\theta(j)$ tangent points $\tau_1, \tau_2, \dots, \tau_{\theta(j)}$ will be located on $\theta(j)$ distinct boundary walls (see Fig. 3c). \square

3.2 Exploration Model

We now show how we can use the preceding definitions and geometric claims to form a model for frontier-based exploration and establish an equivalence between the the medial axis configuration space and the physical environment.

We claim that reasoning about a single point moving along the skeleton allows us to reason about a group of robots that form a frontier with their end to end sensor footprints moving through physical space. We call this point the *swarm locus*. By definition a corridor point $x \in S$ has exactly two tangent points. For a swarm locus stationed at x we call these two points the frontier *anchor points*. The frontier of a group of robots is represented by a corresponding frontier formed by two line segments joining the swarm locus to its anchor points. A group of robots engaged in non-recontaminating exploration will form a frontier arc subtended between two obstacle boundary walls in physical space; the frontier arc separates the inspected region on one side from the contaminated region on the other side. Our abstraction allows us to reason in similar terms: a swarm locus stationed at a point x on the corridor of the skeleton will similarly form a frontier arc consisting of two line segments subtended between two obstacle boundary walls; the frontier arc transposed onto Q likewise separates the inspected region from the contaminated region (see Fig. 4a). We understand the frontier of a swarm locus to mean the frontier $\mathcal{F}' \subseteq \mathcal{F}$ of a group of robots represented by a swarm locus stationed at a point $x \in S$.

Note that we are making a simplifying abstraction in representing the frontier $\mathcal{F}' \subseteq \mathcal{F}$ of a group of n_0 robots by two end to end line segments subtended between two obstacle boundary walls. Observe that as n grows, the abstraction becomes more accu-

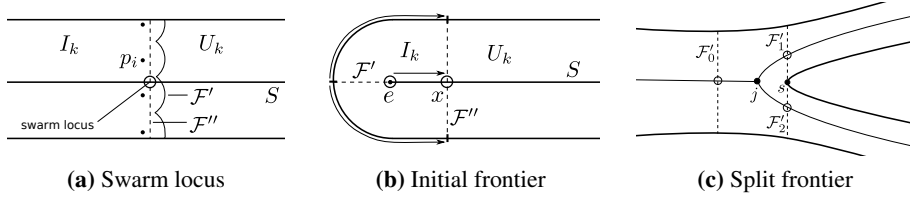


Fig. 4: Fig. 4a shows a group of robots at positions p_i forming a frontier \mathcal{F}' with the exterior boundary of their sensor footprints. Superimposed is the corresponding swarm locus stationed at a point $x \in S$ forming a frontier \mathcal{F}'' with two line segments subtended between two obstacle boundary walls. Fig. 4b shows the initial frontier \mathcal{F}' of a swarm locus stationed at an end point e , separating the environment into $I_0 = \emptyset$ and $U_0 = Q$. As the swarm locus moves along the skeleton to reach a point $x \in S$, it forms a frontier \mathcal{F}'' . The swarm locus has swept across the environment and cleared the region to the left of \mathcal{F}'' . Fig. 4c shows the configuration of a split frontier. A swarm locus is traversing a junction point j with $\theta(j) = 3$. The ingoing frontier \mathcal{F}'_0 splits, producing 1 split point s and 2 outgoing frontiers $\mathcal{F}'_1, \mathcal{F}'_2$.

rate as the periodic protrusion of the frontier due to the curvature of sensor footprints becomes finer-grained and less prominent with respect to its length. In general, this abstraction is justified as we are usually interested in characterizing environments where $n \gg 0$.

For the purposes of introducing the exploration model we assume that the swarm locus always begins at an end point. (Note that this assumption only serves to simplify the discussion, and can be removed easily by introducing several special cases.) From the definition of the degree of a point on the skeleton, a maximal inscribed circle C centered at an end point $e \in S$ will be tangent to the obstacle boundary at a single point $\tau(e)$. Thus both anchor points are the same point $\tau(e)$ and the frontier $\mathcal{F}' \subseteq \mathcal{F}$ of a swarm locus stationed at e is formed by two identical line segments $[e \tau(e)]$. In this configuration, \mathcal{F}' separates the environment Q into the inspected region $I_0 = \emptyset$ and the contaminated region $U_0 = Q$, corresponding to the fact that the swarm locus has not yet explored any of the environment. As the swarm locus starts moving along the skeleton, the anchor points will move along ∂B on either side of the corridor and the frontier will “sweep” across the environment. The frontier now separates Q into two disjoint nonempty regions. The inspected region begins growing, while the contaminated region begins shrinking, corresponding to the fact that the robots have begun clearing the environment (see Fig. 4b).

Moving Through Corridors We define the *relief map* $\mathcal{R} : \mathbb{R}^2 \rightarrow \mathbb{N}$ as the quantization of the distance transform using the sensing radius r , given by $\mathcal{R}(x) = \lceil \mathcal{D}(x)/r \rceil$ (see Fig. 2d). The relief map uses the distance transform to similarly capture the notion of width, expressing the same information in terms of the number of robots required at a point x to reach the closest point on the obstacle boundary.

Lemma 2. *A group of n_0 robots represented by a swarm locus that reaches a corridor point $x \in S$ prevents recontamination if and only if $n_0 \geq \mathcal{R}(x)$.*

Proof. In order to prevent recontamination, the group of robots must subtend a frontier between two obstacle boundary walls in physical space. By definition, the distance between $x \in S$ and the closest point to the obstacle boundary on either side is exactly $\mathcal{D}(x) \leq \mathcal{R}(x)$. Thus we can always produce two lines l_1, l_2 from x to the two closest points on the obstacle boundary of combined length $L \leq 2\mathcal{R}(x)$. If $n_0 \geq \mathcal{R}(x)$ then the robots can always align themselves in physical space such that their frontier passes through x in the same arrangement as l_1, l_2 of combined length $L \geq 2\mathcal{R}(x)$. Therefore a group of $n_0 \geq \mathcal{R}(x)$ robots can always form a frontier that subtends between two obstacle boundary walls, and recontamination can always be prevented. If $n_0 < \mathcal{R}(x)$ then any arrangement of the robots in physical space such that their frontier passes through x will always result in a frontier of length $L < 2\mathcal{R}(x)$. Therefore a group of $n_0 < \mathcal{R}(x)$ robots can never form a frontier that subtends between two obstacle boundary walls, and recontamination will always occur. \square

Corollary 1. *A group of n_0 robots represented by a swarm locus moving along a corridor $G = [a b] \subset S$ prevents recontamination at all points $x \in G$ if and only if $n_0 \geq \max_{x \in G} \{\mathcal{R}(x)\}$.*

When a swarm locus reaches an end point, the situation is the reverse of that at the beginning. From the definition of the degree of a point on the skeleton, a maximal inscribed circle C centered at an end point $e \in S$ will be tangent to the obstacle boundary at a single point $\tau(e)$. Thus both anchor points are the same point $\tau(e)$ and the frontier $\mathcal{F}' \subseteq \mathcal{F}$ of a swarm locus stationed at e is formed by two identical line segments $[e \tau(e)]$. In this configuration, \mathcal{F}' separates the environment Q into the inspected region I_t and some part of the contaminated region \emptyset , corresponding to the fact that the swarm locus has cleared a particular corridor. In the case where there is only one swarm locus, \mathcal{F}' separates the environment Q into the inspected region $I_t = Q$ and the entire contaminated region $U_t = \emptyset$, corresponding to the fact that the entire environment has been cleared.

Traversing Junctions When a group of robots reaches a junction in physical space, they should split and explore the outgoing junction corridors separately. If the robots do not split then recontamination will occur due to any of the unattended outgoing corridors coming in contact with the inspected region. Upon reaching a physical junction with some number of outgoing corridors, a single group of robots forms more than one group of robots with that number of disjoint maximally connected arcs making up the exterior boundary of their sensor footprints. Correspondingly, we define the frontier $\mathcal{F}'_0 \subseteq \mathcal{F}$ of a group of robots to *split* when the exterior boundary of the sensor footprints of the robots is no longer connected and becomes the union of two or more disjoint maximally connected arcs. Thus a group of robots splits when their frontier splits. For a junction point j with $\theta(j) - 1$ outgoing corridors, the junction is considered *traversed* when the frontier $\mathcal{F}'_0 \subseteq \mathcal{F}$ of the group of robots splits to form $\theta(j) - 1$ outgoing frontiers $\mathcal{F}'_1, \mathcal{F}'_2, \dots, \mathcal{F}'_{\theta(j)-1} \subset \mathcal{F}$.

Observe that since the total frontier is at all times given by $\mathcal{F} = \partial I \setminus \partial I_B$, if one or more new disjoint maximally connected frontier arcs form, then by necessity one or more new disjoint maximally connected inspected obstacle boundary arcs also form. At the time t_0 that a frontier $\mathcal{F}'_0 \subseteq \mathcal{F}$ of a group of robots splits to form $\theta(j) - 1$ frontiers,

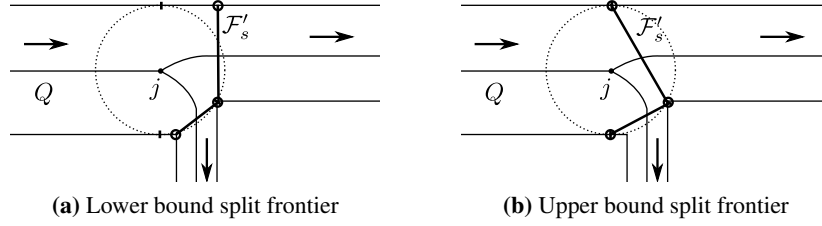


Fig. 5: Frontier configurations for lower and upper bound derivations.

it will have $\theta(j) - 2$ points of contact with the obstacle boundary. We call a frontier \mathcal{F}'_0 in such a configuration a *split frontier* and we call these points *split points* (see Fig. 4c). For $t > t_0$ the split points on the obstacle boundary grow into the $\theta(j) - 2$ new disjoint maximally connected inspected obstacle boundary arcs.

Split Frontier Bounds We establish the lower bound as follows. To traverse a junction we require a split frontier \mathcal{F}'_s to be formed with $\theta(j) - 2$ split points. The ingoing frontier \mathcal{F}'_0 subtends between two boundary walls $\partial B_1, \partial B_{\theta(j)}$, intersecting them at the frontier anchor points c_1, c_2 . Therefore the split points $s_1, s_2, \dots, s_{\theta(j)-2}$ are located on each of the other $\theta(j) - 2$ boundary walls $\partial B_2, \partial B_3, \dots, \partial B_{\theta(j)-1}$. Without a split point on each of these boundary walls, the split frontier cannot be formed and the junction cannot be traversed. Hence, a lower bound on the number of robots required to traverse a junction is given by minimizing the length of the split frontier over all possible points on the respective boundary walls. Thus we can never form a split frontier of total length less than

$$\|\mathcal{F}'_{s,min}\| = \min \left\{ \|c_1 s_1\| + \|s_1 s_2\| + \dots + \|s_{\theta(j)-3} s_{\theta(j)-2}\| + \|s_{\theta(j)-2} c_2\| \right\},$$

for $c_1 \in \partial B_1, s_1 \in \partial B_2, \dots, s_{\theta(j)-2} \in \partial B_{\theta(j)-1}, c_2 \in \partial B_{\theta(j)}$. (1)

If the number of robots is insufficient to form a split frontier of the smallest possible length, then it will certainly be insufficient to form a split frontier of any other length. But a split frontier must be formed in order to traverse a junction, irrespective of the exploration model. Thus no exploration model can dictate a configuration of robots that is able to traverse a given junction with fewer than $\|\mathcal{F}'_{s,min}\|$ robots. This establishes the lower bound (see Fig. 5a).

We establish the upper bound as follows. Assume we have some number of robots at a junction of degree $\theta(j)$. The ingoing frontier \mathcal{F}'_0 subtends between two boundary walls $\partial B_1, \partial B_2$, intersecting them at the frontier anchor points c_1, c_2 . The ingoing frontier can always be aligned, without admitting recontamination, such that its anchor points correspond to the tangent points $\tau_1, \tau_{\theta(j)}$ on the two boundary walls that it subtends. By Lemma 1, a junction of degree $\theta(j)$ will have tangent points on $\theta(j)$ distinct boundary walls. Thereafter the ingoing frontier can always be extruded toward each of the other $\theta(j) - 2$ tangent points in turn, likewise without admitting recontamination. Thus for

a junction j with tangent points $\tau_1, \tau_2, \dots, \tau_{\theta(j)}$ we can always form a split frontier of total length

$$\begin{aligned} \|\mathcal{F}'_{s,max}\| &= \|c_1 s_1\| + \|s_1 s_2\| + \dots + \|s_{\theta(j)-3} s_{\theta(j)-2}\| + \|s_{\theta(j)-2} c_2\| \\ &= \|\tau_1 \tau_2\| + \|\tau_2 \tau_3\| + \dots + \|\tau_{\theta(j)-1} \tau_{\theta(j)}\| . \end{aligned} \quad (2)$$

We can traverse any junction in this way, thus no junction will ever require more than $\|\mathcal{F}'_{s,max}\|$ robots to traverse. This establishes the upper bound (see Fig. 5b). Observe that since all the tangent points are on the boundary of a maximal inscribed circle centered at j , the ingoing frontier \mathcal{F}'_0 , aligned such that its anchor points correspond to $\tau_1, \tau_{\theta(j)}$, is at most $2\mathcal{D}(j)$ in length, i.e. the diameter of the circle. For each of the $\theta(j) - 2$ split points, the frontier gains an additional line segment, likewise of at most $2\mathcal{D}(j)$ in length. Thus we get a numeric upper bound on the maximum length of the split frontier, given by

$$\|\mathcal{F}'_{s,max}\| \leq 2(\theta(j) - 1)\mathcal{D}(j) . \quad (3)$$

We have now established an equivalence between the medial axis configuration space and the physical movement and frontier expansion of robots in physical space. We introduced an exploration model whereby a swarm locus moving along the skeleton allows us to reason about a group of robots moving through physical space. We defined what it means for a frontier to split and for a group of robots to traverse a junction, and derived lower and upper bounds on the number of robots required to traverse a junction.

4 Topology Tree

In this section we present a series of steps that will transform our continuous configuration space into a symbolic representation of the environment in the discrete domain. We establish a set of rules for navigating the environment in this discrete representation, that allow us to develop an algorithmic pursuit strategy. Using the junction lower bound from Result (1) we derive a lower bound on the total number of pursuers necessary to clear the environment, showing that no fewer than this number can possibly clear the environment regardless of the exploration model or pursuit strategy. Using the junction upper bound from Result (2) we develop an upper bound on the total number of pursuers that will always be sufficient to clear the environment, for any pursuit strategy. Finally, we derive an optimal pursuit strategy and prove that it guarantees we can clear the environment with the minimum number of pursuers for a given exploration model.

The most natural representation of the skeleton of an environment where both Q and B are connected is a tree. Since we are also given a starting point on the skeleton, we consider a directed rooted tree (rooted at the start node). We refer to this as the environment *topology tree*, denoted by $T = (V, E)$, where V is the set of vertices (nodes) and E is the set of edges of T . Let $s \in V$ be the root node of T . Nodes correspond to end points and junction points, and edges correspond to corridors connecting these points on the skeleton.

Let $\gamma : V \rightarrow \mathbb{N}$ denote the out-degree of a node. There are four types of nodes on the topology tree. The swarm locus starts at an end point on the skeleton which corresponds

to the root node s of out-degree $\gamma(s) = 1$. Every other end point corresponds to a leaf node w of out-degree $\gamma(w) = 0$. Each junction point j is represented by a unique junction *entry node* u of out-degree $\gamma(u) = \theta(j) - 1$, connected to an associated set of distinct junction *exit nodes* $\{v_1, v_2, \dots, v_{\gamma(u)}\}$ of out-degree $\gamma(v) = 1$. (Observe that since T is a directed rooted tree, every node has in-degree 1, except for the root node s which has in-degree 0.)

The root node is connected to some junction entry node, while junction exit nodes are connected to leaf nodes and other junction entry nodes, as determined by the corridors connecting these points on the skeleton. Every edge $e \in E$ on the topology tree is assigned a weight $\alpha : E \rightarrow \mathbb{N}$. There are two fundamentally different types of edges: edges that represent a corridor on the skeleton and edges that represent the split frontier at a junction point.

Motivated by Corollary 1, for an edge e connecting node u of out-degree $\gamma(u) = 1$ to a node v , the edge weight $\alpha(e)$ is determined by the number of robots necessary and sufficient to advance from u to v , given by the maximum relief value $\mathcal{R}(x)$ at some point $x \in S$ amongst the points along that corridor.

Motivated by the reasoning in Section 3.2.2, for junction nodes the representation is as follows. For each junction entry node $u \in V$ let $E_u \subset E$ be the set of edges $\{e_1, e_2, \dots, e_{\gamma(u)}\}$ connecting u to its associated set of junction exit nodes $\{v_1, v_2, \dots, v_{\gamma(u)}\}$. For each junction, we define a traversal function $\delta : E_u \rightarrow \mathbb{N}$, where $\sum_i \delta(e_i)$ is the number of robots required to traverse the junction. This corresponds to the minimum number of robots required to form a split frontier at the junction, given by its ceiling length $\lceil \|\mathcal{F}'_s\| \rceil$. Since we do not have tight bounds on the length of the split frontier, the traversal function depends on the context of the analysis. Namely, if the goal is to derive a lower bound on the number of robots required to clear an environment, then we consider the length of the split frontier $\lceil \|\mathcal{F}'_{s,min}\| \rceil$ given by Result (1). If the goal is to derive an upper bound then we consider the length of the split frontier $\lceil \|\mathcal{F}'_{s,max}\| \rceil$ given by Result (2). Each edge e_i is assigned weight $\alpha(e_i) = \delta(e_i)$ which corresponds to the number of robots n_i that are required to form a frontier \mathcal{F}'_i at each outgoing corridor, given by the ceiling length of each outgoing frontier $\lceil \|\mathcal{F}'_i\| \rceil$.

4.1 Exploration Rules

We consider exploration of the topology tree to be a game. We start the game with a single group of n_0 robots stationed at the root node s . Every node $v \in V$ on the topology tree is marked with a label λ , which can have one of three values: CONTAMINATED, EXPLORED and CLEARED. Initially, the root node is marked EXPLORED and all other nodes are marked CONTAMINATED.

We play the game in rounds, each round moving some number of robots from one node to another. If a group of robots is unable to move from one node to another on some round, then the robots are “stuck” at that node. This corresponds to the fact that if there are insufficient robots to clear a corridor, they will remain stuck guarding the corridor, unable to retreat without allowing recontamination. Let $\lambda_k(v)$ denote the labeling of a node $v \in V$ on round k . We win the game if the tree is cleared on some round k_0 , that is if $\exists k_0 \in \mathbb{N} \mid \forall k > k_0 \forall v \in V, \lambda_k(v) = \text{CLEARED}$. We lose the game if all robots are stuck at some node but the tree has not been cleared by some round k_0 , that is if $\exists k_0 \in \mathbb{N} \mid \forall k > k_0 \exists v \in V, \lambda_k(v) \neq \text{CLEARED}$.

Robots can split into smaller groups and join to form larger groups. In general, we are free to choose how we move the robots on the topology tree, provided that we obey the following transition rules.

1. If a group of n_0 robots reaches a node u where $\gamma(u) > 0$, then the group splits into some permutation of $\gamma(u)$ groups of n_i robots advancing to each of the children nodes $\{v_1, v_2, \dots, v_{\gamma(u)}\}$. We are free to choose this permutation, subject to the following restrictions:
 - (a) If $\lambda(v_i) = \text{CONTAMINATED}$, then $n_i \geq \alpha(e_i(u))$.
 - (b) If $\lambda(v_i) = \text{EXPLORED}$, then $n_i \geq 0$.
 - (c) If $\lambda(v_i) = \text{CLEARED}$, then $n_i = 0$.
 If no such permutation exists, then the group remains stuck at node u .
2. If a group of robots is stationed at a node u where $\lambda(u) = \text{CLEARED}$, then the group backtracks to the parent node.
3. If a group of robots reaches a node u that is marked **CONTAMINATED**, then u is marked **EXPLORED**.
4. If a group of robots reaches a leaf node u or a node u where all children of u are marked **CLEARED**, then u is also marked **CLEARED**.
5. If two or more groups of n_1, n_2, \dots, n_k groups of robots are stationed at the same node, then they form a single group of $n_1 + n_2 + \dots + n_k$ robots.

The reasoning behind these rules follows from the problem formulation and results in Section 3. Rule 1(a) enforces that our exploration is non-recontaminating. Rule 1(b) allows robots to move to explored nodes and join other robots. Rules 3 and 4 define the progression of the game, and Rules 1(c) and 2 ensure that exploration is always progressive (the latter ensures a group of robots leaves a region once it has been cleared, while the former ensures that no group of robots re-enters that region unnecessarily). Rule 5 ensures that robots always act in a single group when stationed at a node.

We understand a state of the topology tree to mean the labeling of each node and the number of robots stationed at each node on a given round. We call a sequence of transitions between states of the topology tree a *pursuit strategy*. (We omit a formal definition for brevity.) A pursuit strategy is like a written record of a game of chess that allows the game to be replayed by carrying out the recorded sequence of transitions. Observe that the only degree of freedom in choosing a pursuit strategy is what to do at a given junction entry node u . We can always choose what junction exit node to send a group of robots to as long as it is not marked **CLEARED**. If the junction exit nodes are marked **CONTAMINATED** then the group traverses the junction if and only if $n_0 \geq \sum_i \alpha(e_i(u))$. If the junction is traversed, then the group splits into some permutation of $\gamma(u)$ groups of n_i robots advancing to each of the associated junction exit nodes $\{v_1, v_2, \dots, v_{\gamma(u)}\}$. We are free to choose this permutation, provided that $\forall i, n_i \geq \alpha(e_i(u))$. The choice we make in selecting this permutation may affect the outcome of the game. (Note also that a node u is only marked **CLEARED** once the entire subtree $T(u)$ is marked cleared. Thus robots are forced to clear subtrees recursively, and can only backtrack once a given subtree is cleared.)

We also note that because n_0 and $|V|$ are finite, there are a finite number of possible pursuit strategies for a given topology tree. Intuition tells us that if n_0 is too low, every

pursuit strategy will be a losing strategy, whereas if n_0 is sufficiently high then any pursuit strategy will be a winning strategy. We now formalize this intuition, and derive lower and upper bounds on the total number of robots that can clear the topology tree.

4.2 Environment Bounds

Consider the topology tree T with root node s . Let $T(q)$ be the tree obtained by considering node q as the root node and removing nodes that are not descendants of q . Let $n(T(q))$ be the number of robots required to clear $T(q)$. Let $P(v)$ be the set of nodes $\{p_1, p_2, \dots, p_{\gamma(v)}\}$ that are children of $v \in V$, enumerated in order of ascending $n(T(p_i))$.

We motivate the lower bound as follows. At each node s we consider whether more robots are required to advance to the child node p_1 than are required to clear the rest of the subtree $T(p_1)$, and apply this recursively for the entire tree. At each junction entry node we consider the maximum number of robots required to clear a given subtree $T(p_i)$ while guarding the remaining junction exit nodes that have not been cleared. Formally, let $n_{min}(T(s))$ be the total number of robots necessary to clear the environment with topology tree T , given by

$$n_{min}(T(s)) = \begin{cases} 0 & \text{if } \gamma(s) = 0 \\ \max \left\{ \sum_{i=1}^{\gamma(s)} \alpha(e_i(s)), \max_{i=1, \dots, \gamma(s)} \left\{ n_{min}(T(p_i)) + \sum_{j=i+1}^{\gamma(s)} \alpha(e_j(s)) \right\} \right\} & \text{otherwise.} \end{cases} \quad (4)$$

Lemma 3. $n_{min}(T(s))$ robots are necessary to clear an environment with topology tree T , regardless of exploration model or pursuit strategy.

Proof. We begin at the root node s , which has out-degree $\gamma(s) = 1$. If $\alpha(e_1(s)) > n(T(p_1))$, i.e. if more robots are required to advance to the child node $p_1 \in P(s)$ than to clear the rest of the tree $T(p_1)$, then $n_{min}(T(s)) = \alpha(e_1(s))$. (The second term in the outer max expression is not evaluated if $\gamma(s) = 1$.) This applies recursively for any node of out-degree $\gamma(s) = 1$. Leaf nodes provide the recursion base case, where $n(T(w)) = 0$ for $w \in V$ if $\gamma(w) = 0$.

For junctions the logic is as follows. For a junction entry node $u \in V$, each associated junction exit node $p_i \in P(u)$ is the root node of a subtree $T(p_i)$. By Rules 1 and 2, a group of robots that reaches a leaf node w will backtrack until it reaches a node with previously unexplored children. By induction a group of robots that clears $T(p_i)$ will backtrack until it returns to u and advance to a different junction exit node $p_{j \neq i}$. By Rule 5, this group of robots will join with any other group of robots stationed at p_j . When a group of robots reaches a junction, at least $n_0 > \sum_i \alpha(e_i(u))$ robots are required to traverse it thus stationing $n_i = \alpha(e_i(u))$ robots at the $\gamma(u)$ junction exit nodes p_i . Thereafter n_0 must be at least enough to clear the subtree with smallest $n(T(p_i))$ while holding station at the remaining $\gamma(u) - 1$ exit nodes. This subtree is $T(p_1)$ since $p_i \in P(u)$ are enumerated in order of ascending $n(T(p_i))$. If n_0 robots is not enough

to clear $T(p_1)$, then it will certainly not be enough to clear any other subtree, and thus the junction cannot be traversed. If n_0 robots is enough to clear $T(p_1)$ then $n(T(p_1))$ robots are now able to join one of the groups guarding the remaining $\gamma(u) - 1$ junction exit nodes. We apply this iteratively for all subtrees $T(p_i)$, each time gaining the services of the group that cleared the previous subtree. The final subtree simply requires $n(T(p_{\gamma(u)}))$ robots, since all other subtrees have been cleared and do not require any robots to guard the junction exit nodes. (Note that all n_i groups of robots will actually clear their subtrees simultaneously, in which case two given groups may not join at the junction exit nodes, but elsewhere along a given subtree. However, considering each group to clear its subtree in stages with the other groups guarding the junction exit nodes simplifies the abstraction.) We now consider the maximum of the number of robots required to traverse the junction (the first term in the outer max expression) and the maximum number of robots required to clear a given subtree $T(p_i)$ and guard the remaining $\gamma(u) - i$ junction exit nodes (the second term in the outer max expression, itself a maximum over $\gamma(u)$ stages). The maximum of these two outer terms is the minimum number of robots required to clear the subtree $T(u)$. We apply this logic recursively for all junctions.

Thus no fewer than $n_{min}(T(s))$ robots can clear an environment with topology tree T , for a given exploration model. Using the junction lower bound from Result (1) to obtain the traversal function δ_{min} for each junction, we know that no fewer than $\sum_i \delta_{min}(e_i(u))$ robots can traverse the junction point j corresponding to the junction entry node u . Thus, using $\alpha(e_i(u)) = \delta_{min}(e_i(u))$ for each junction entry point u , $n_{min}(T(s))$ gives a lower bound on the number of robots that is necessary to clear an environment with topology tree T , regardless of exploration model or pursuit strategy. \square

We motivate the upper bound as follows. We imagine an adversary that dictates the pursuit strategy of a number of robots, with the goal of placing the maximum number of them on the topology tree T , while preventing T from being cleared. Then we argue that given any such adversarial configuration of $n^*(T(s))$ robots, $n^*(T(s)) + 1$ robots will always be able to clear T , regardless of the pursuit strategy chosen by the adversary. Formally, let $n_{max}(T(s))$ be the total number of robots sufficient to clear the environment with topology tree T , given by

$$n_{max}(T(s)) = n^*(T(s)) + 1, \quad \text{if } \gamma(s) = 0$$

$$n^*(T(s)) = \begin{cases} 0 & \text{if } \gamma(s) = 0 \\ \max \left\{ \left(\sum_{i=1}^{\gamma(s)} \alpha(e_i(s)) \right) - 1, \sum_{i=1}^{\gamma(s)} n^*(T(p_i)) \right\} & \text{otherwise.} \end{cases} \quad (5)$$

Lemma 4. $n_{max}(T(s))$ robots are sufficient to clear an environment with topology tree T , for a given exploration model, regardless of pursuit strategy.

Proof. The adversary begins at the root node s , which has out-degree $\gamma(s) = 1$. If $\alpha(e_1(s)) > n(T(p_1))$, i.e. if more robots are required to advance to the child node $p_1 \in P(s)$ than to clear the rest of the tree $T(p_1)$, then $n^*(T(s))$ simply equals $\alpha(e_1(s)) - 1$

since adding one more robot will result in the entire tree being cleared. This applies recursively for any node of out-degree $\gamma(s) = 1$. Leaf nodes provide the recursion base case, where $n(T(w)) = 0$ for $w \in V$ if $\gamma(w) = 0$.

For junctions the logic is as follows. For a junction entry node $u \in V$, each associated junction exit node $p_i \in P(u)$ is the root node of a subtree $T(p_i)$. The adversary can choose one of two options: either place $(\sum_{i=1}^{\gamma(s)} \alpha(e_i(s))) - 1$ robots at u such that the junction cannot be traversed, or traverse the current junction with $\sum_{i=1}^{\gamma(s)} n^*(T(p_i))$ robots knowing that they will not be able to clear the rest of the tree. The adversary chooses the maximum of these two values since she is trying to maximize the number of robots on the tree. We apply this logic recursively for all junctions.

Consider any such configuration of $n^*(T(s))$ robots. We now introduce one additional robot at the root node s . The robot must navigate the tree according to the exploration rules, but the adversary is still free to choose its pursuit strategy. By Rule 5, whenever the robot reaches any node u with $n^*(T(u))$ robots stationed at the node, a single group of $n^*(T(u)) + 1$ robots forms at u which is sufficient to clear the subtree $T(u)$. By Rule 2, any group of robots that clears a subtree will backtrack along the tree until it reaches a previously unexplored part of the tree. This group of robots will join other groups of $n^*(T(v))$ robots similarly stationed at other nodes $v \in V$. This will continue recursively until the entire tree is cleared. Thus placing an additional robot at s causes a ‘‘chain reaction’’ that results in T being cleared regardless of the pursuit strategy that the adversary chooses for any of the $n^*(T(s)) + 1$ robots.

Thus no more than $n_{max}(T(s)) = n^*(T(s)) + 1$ robots will ever be required to clear an environment with topology tree T , regardless of pursuit strategy. Using the junction upper bound from Result (2) to obtain the traversal function δ_{max} for each junction, we know that no more than $\delta_{max}(e_i(u))$ robots are required to traverse the junction point j corresponding to the junction entry node u for the given exploration model. Thus, using $\alpha(e_i(u)) = \delta_{max}(e_i(u))$ for each junction entry point u , $n_{max}(T(s))$ gives an upper bound on the number of robots that is sufficient to clear an environment with topology tree T , for a given exploration model, regardless of pursuit strategy. \square

4.3 Optimal Pursuit Strategy

We now present an optimal pursuit strategy that guarantees that the environment is cleared with the minimum number of robots for a given exploration model. Consider the topology tree T with root node s . We know that $n_{min}(T(s))$ robots are necessary to clear T , given by Result (4). The following algorithm guarantees that T will be cleared with $n_{min}(T(s))$ robots. (We use the same notation as in Section 4.2.)

Algorithm 1 – Clear($T(s)$, n_0)

Given n_0 robots located at root node s of topology tree $T = (V, E)$:

1. If s is a leaf node or if all children of s are marked CLEARED, then:
 - (a) mark $s \leftarrow$ CLEARED.
 - (b) Backtrack to parent node. If parent node does not exist, terminate.
2. If $\lambda(s) =$ CONTAMINATED, mark $s \leftarrow$ EXPLORED.
3. If $\gamma(s) = 1$, Clear($T(p_1)$, n_0).

4. If $\gamma(s) > 1$ and if all children of s are marked CONTAMINATED, then:
 - (a) for $i = 2, \dots, \gamma(s)$: $\text{Clear}(T(p_i), \alpha(e_i(s)))$.
 - (b) $\text{Clear}(T(p_1), n_0 - \sum_{i=2}^{\gamma(s)} \alpha(e_i(s)))$.
5. If $\gamma(s) > 1$ and if all children of s are not marked CONTAMINATED, then:
 - (a) let $i = \min_{2, \dots, \gamma(s)} \{i \mid \lambda(p_i) \neq \text{CLEARED}\}$.
 - (b) $\text{Clear}(T(p_i), n_0)$.

Lemma 5. $n_{\min}(T(s))$ robots are necessary and sufficient to clear an environment with topology tree T , for a given exploration model.

Proof. Given $n_{\min}(T(s))$ robots we prove that we can use the `Clear` algorithm to clear the environment. Let $n_0 = n_{\min}(T(s))$. We begin at the root node s , which has out-degree $\gamma(s) = 1$. A group of robots can always advance to the child node p_1 since we have $n_0 = n_{\min}(T(s))$ robots. This applies for all nodes of out-degree $\gamma(s) = 1$. When a group of robots reaches a leaf node it is marked CLEARED. A group of robots stationed at a node that is marked CLEARED will backtrack until it reaches a node with children that are not marked CLEARED.

For junctions, the logic is as follows. The first time a group of robots reaches a given junction entry node u , all children of u are marked CONTAMINATED. We send $n_i = \alpha(e_i(s))$ robots to each of the $\gamma(s) - 1$ junction exit nodes p_i , and use the remaining $n_1 = n_0 - \sum_{i=2}^{\gamma(s)} \alpha(e_i(s))$ robots to clear the subtree $T(p_1)$. By proof to Lemma 3, we know that the entire subtree $T(p_1)$ can be cleared with n_1 robots, and therefore n_1 robots will eventually backtrack to u . Thereafter, all children of u are either EXPLORED or CLEARED. Each group of robots reaching u in this way is sent to clear $T(p_i)$ where i is the smallest number such that $\lambda(p_i) \neq \text{CLEARED}$. We do this iteratively for each subtree, each time gaining the services of the group that cleared the previous subtree. We can always clear each subtree $T(p_i)$ in this manner because $\{p_1, p_2, \dots, p_{\gamma(s)}\}$ are enumerated in order of ascending $n(T(p_i))$, and by proof to Lemma 3 we know that $n(T(u))$ robots is enough to clear every subtree in this way. For each subtree $T(p_i)$ cleared in this way, p_i is marked CLEARED. When all subtrees have been cleared, u will be marked CLEARED, and the entire group of robots will backtrack to the parent node. This applies recursively recursively for all junctions. \square

Using the junction upper bound from Result (2) to obtain the traversal function δ_{\max} for each junction, we know that no more than $\delta_{\max}(e_i(u))$ robots are required to traverse the junction point j corresponding to the junction entry node u for the given exploration model. Thus, using $\alpha(e_i(u)) = \delta_{\max}(e_i(u))$ for each junction entry point u , the `Clear` algorithm gives an optimal pursuit strategy for clearing an environment with topology tree T , for a given exploration model.

5 Conclusion

The problem of obtaining a concise characterization of a physical environment in the context of frontier-based non-recontaminating exploration was considered. We introduced the medial axis as a configuration space and showed that reasoning about points in this configuration space is equivalent to reasoning about robots in physical space.

We formalized the notion of width, corridors and junctions and derived lower and upper bounds on the number of robots required to traverse a junction. We presented a transformation from this continuous configuration space into a symbolic representation in the discrete domain. We cast the exploration problem as a game, established rules for playing this game, and derived bounds on the number of robots necessary and sufficient to clear the environment. Finally we presented an optimal pursuit strategy that guarantees that we can clear the environment with the minimum number of robots.

There are a number of interesting future lines of research for this work. First, the establishment of tight bounds on the number of robots required to traverse a junction — we suspect that the lower bound given by Result (1) in Section 3.2 is in-fact necessary and sufficient. A rigorous proof of this fact would have to generalize to accommodate a number of special cases.

Second, the extension of this work to environments with holes would be a significant contribution. The medial axis configuration space was chosen with this in mind, and the model presented in Section 3 soundly generalizes the characterization to arbitrary connected environments. A number of issues need to be addressed in transforming this representation into the discrete domain. One would need to consider an undirected graph and junctions would need to be represented accordingly. It is known that computing the number of searchers required to clear a general graph is NP-hard [21], so suitable heuristics or approximations would need to be employed. Alternatively, the graph could be converted into a tree such as in [15], [16]; however the non-isotropic nature of the junction transition function would demand a judicious approach to blocking cycles.

Acknowledgements This research was supported in part by the Future Urban Mobility project of the Singapore-MIT Alliance for Research and Technology (SMART) Center, with funding from Singapore’s National Research Foundation. It was also supported in part by AFOSR Award Number FA9550-08-1-0159, NSF Award Number CNS-0715397, NSF Award Number CCF-0726514, ONR grant N000140911051 and NSF grant 0735953. We are grateful for this support.

References

1. M. Adler, H. Racke, N. Sivadasan, C. Sohler, and B. Vocking. Randomized pursuit-evasion in graphs. *Combinatorics, Probability and Computing*, 12(03):225–244, 2003.
2. H. Blum. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, 19:362–380, 1967.
3. J. Burgos. Pursuit/evasion behaviors for multi-agent systems. 2010.
4. H. Choi, S. Choi, and H. Moon. Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 181(1):57–88, 1997.
5. P. Danielsson. Euclidean distance mapping. *Computer Graphics and image processing*, 14(3):227–248, 1980.
6. P. Dimitrov, C. Phillips, and K. Siddiqi. Robust and efficient skeletal graphs. In *cvpr*, page 1417. Published by the IEEE Computer Society, 2000.
7. J. Durham, A. Franchi, and F. Bullo. Distributed pursuit-evasion with limited-visibility sensors via frontier-based exploration. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3562–3568. IEEE, 2010.
8. B. Gerkey, S. Thrun, and G. Gordon. Visibility-based pursuit-evasion with limited field of view. *The International Journal of Robotics Research*, 25(4):299, 2006.

9. P. J. Giblin and B. B. Kimia. Local forms and transitions of the medial axis. In K. Siddiqi and S. M. Pizer, editors, *Medial Representations*, volume 37 of *Computational Imaging and Vision*, pages 37–68. Springer Netherlands, 2008.
10. L. Guibas, C. Holleman, and L. Kavraki. A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach. In *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 1, pages 254–259. IEEE, 1999.
11. L. Guibas, J. Latombe, S. Lavallo, D. Lin, and R. Motwani. Visibility-based pursuit-evasion in a polygonal environment. *Algorithms and Data Structures*, 1272:17–30, 1997.
12. L. Guibas, J. Latombe, S. LaValle, D. Lin, and R. Motwani. A visibility-based pursuit-evasion problem. *International Journal of Computational Geometry and Applications*, 9(4/5):471, 1999.
13. C. Holleman and L. Kavraki. A framework for using the workspace medial axis in prm planners. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1408–1413. IEEE, 2000.
14. V. Isler, D. Sun, and S. Sastry. Roadmap based pursuit-evasion and collision avoidance. In *Proc. Robotics, Systems, & Science*, 2005.
15. A. Kolling and S. Carpin. The graph-clear problem: definition, theoretical properties and its connections to multirobot aided surveillance. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1003–1008. IEEE, 2007.
16. A. Kolling and S. Carpin. Multi-robot surveillance: an improved algorithm for the graph-clear problem. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2360–2365. IEEE, 2008.
17. A. Kolling and S. Carpin. Pursuit-evasion on trees by robot teams. *Robotics, IEEE Transactions on*, 26(1):32–47, 2010.
18. A. Kuijper. Deriving the medial axis with geometrical arguments for planar shapes. *Pattern Recognition Letters*, 28(15):2011–2018, 2007.
19. S. LaValle, D. Lin, L. Guibas, J. Latombe, and R. Motwani. Finding an unpredictable target in a workspace with obstacles. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 737–742. IEEE, 1997.
20. R. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43(2-3):235–239, 1983.
21. T. Parsons. Pursuit-evasion in a graph. *Theory and applications of graphs*, 642:426–441, 1978.
22. S. Rodriguez, J. Denny, T. Zourntos, and N. Amato. Toward simulating realistic pursuit-evasion using a roadmap-based approach. *Motion in Games*, 6459:82–93, 2010.
23. S. Sachs, S. LaValle, and S. Rajko. Visibility-based pursuit-evasion in an unknown planar environment. *The International Journal of Robotics Research*, 23(1):3, 2004.
24. J. Serra. *Image analysis and mathematical morphology*. London.: Academic Press.[Review by Fensen, EB in: J. Microsc. 131 (1983) 258.] Technique Staining Microscopy, Review article General article, Mathematics, Cell size (PMBD, 185707888), 1982.
25. I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal on computing*, 21:863, 1992.
26. S. Wilmarth, N. Amato, and P. Stiller. Maprm: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1024–1031. IEEE, 1999.
27. M. Yamashita, H. Umemoto, I. Suzuki, and T. Kameda. Searching for mobile intruders in a polygonal region by a group of mobile searchers. *Algorithmica*, 31(2):208–236, 2001.
28. B. Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE, 1997.

29. B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on Autonomous agents*, pages 47–53. ACM, 1998.