Optimal Robot Localization in Trees

Rudolf Fleischer

Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong
E-mail: rudolf@cs.ust.hk

Kathleen Romanik[1]

Powerize.com, 901 Elkridge Landing Road, Suite 350, Linthicum, Maryland 21090

Sven Schuierer[2]

Institut für Informatik, Universität Freiburg, Am Flughafen 17, D-79110 Freiburg, Germany
E-mail: sven.schuierer@pharma.Novartis.com

and

Gerhard Trippen[3]

Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong
E-mail: trippen@cs.ust.hk

Received September 8, 1999

The problem of localization, that is, of a robot finding its position on a map, is an important task for autonomous mobile robots. It has applications in numerous areas of robotics ranging from aerial photography to autonomous vehicle exploration. In this paper we present a new strategy LPS (Localize-by-Placement-Separation) for a robot to find its position on a map, where the map is represented as a geometric tree of bounded degree. Our strategy exploits to a high degree the self-similarities that may occur in the environment. We use the framework of competitive analysis to analyze the performance of our strategy. In particular, we show that the distance traveled by the robot is at most $O(\sqrt{n})$ times longer than the shortest possible route to localize the robot, where n is the number of vertices of the tree. This is a significant improvement over the best known previous bound of $O(n^{2/3})$. Moreover, since there is a lower bound of $\Omega(\sqrt{n})$, our strategy is optimal up to a constant factor. Using the same approach we can also show that the problem of searching for a target in a geometric tree, where the robot is given a map of the tree and the location of the target but does not know its own position, can be solved by a strategy with a competitive ratio of $O(\sqrt{n})$, which is again optimal up to a constant factor. © 2001 Elsevier Science

Key Words: tree localization; competitive analysis; online algorithm.

1. INTRODUCTION

In many tasks of autonomous mobile robots it is assumed that the robot has a map of its environment and knows its location on the map. However, in some situations the robot may not know in advance where its correct position on the map is and has to determine it on-line. This is called the *robot localization problem*. Usually it is assumed that this problem can be solved by using sensor data and allowing the robot to move only a small amount. But if the environment consists of many self-similar parts, this approach may not be sufficient.

Although in robotics this issue has been addressed in numerous contexts, ranging from aerial photography to autonomous vehicles for the exploration of landscapes [13, 16, 18, 19], a more rigorous analysis of the problem in a well-defined theoretical framework has only been considered very recently. Here, the environment of the robot is assumed to be a simple or multiply connected polygon P and the robot is assumed to have access to its local visibility polygon V, i.e., all the points that are visible from its position via a range sensing device (for example, a sonar or a ladar). The robot is also assumed to

[1] Supported by AFOSR Grant F49620-93-1-0039.

[2] Supported by the DFG-Project "Diskrete Probleme," Ot 64/8-2.

[3] Supported by UGC Grant AOE98/99.EG01.

have a compass so that it knows its orientation. The first task of the robot is to determine its possible placements within P, that is, all $p \in P$ such that the visibility polygon of p equals V. Guibas *et al.* provide a data structure that allows efficient enumeration of all such positions [9].

If more than one placement exists, then the robot has to travel to certain points of the polygon that allow it to distinguish between the different placements. Of course, it is always possible to find a path that uniquely identifies the true position of the robot; for instance, if the robot follows the boundary of P and the holes of P, then all interior points are visible at some time to the robot and it can localize easily. However, the robot should not travel much farther than necessary. In order to measure the performance of a localization strategy we employ the framework of competitive analysis [17]. If $L(p, P)$ is the length of a shortest path to localize a robot "waking" in polygon P at position p, a strategy S is called *c-competitive* if the length of the path traveled by a robot using strategy S is at most c times $L(p, P)$, for all possible polygons P and points $p \in P$. The value c is called the *competitive ratio* of S.

There have been two approaches to the localization problem. The first by Dudek, Romanik, and Whitesides [7] (see also [15]) considers the full complexity of the problem and utilizes a decomposition of the polygon P into visibility cells such that the same set of vertices of P is visible from each point in a cell. This decomposition is used as the underlying structure for a simple strategy where the robot repeatedly travels to the closest point that eliminates at least one of the possible placements. It is easy to show that if there are k possible placements, then the strategy is k-competitive (Corollary 4 in [7], Lemma 7 in [11]). Furthermore, if $k \leq \sqrt{n}$, where n is the number of vertices of the polygon, then it can be shown that k is the best competitive ratio possible [7].

The second approach, proposed by Kleinberg, leaves aside all concerns raised by the visibility structure of P and abstracts the combinatorial nature of the problem [11]. In this context two types of environments are considered: undirected bounded-degree trees embedded in the d-dimensional Euclidean space $\mathbb{E}^d$ (called geometric trees) and rectangle packings in the plane. In these environments the robot has no use of vision other than to know the orientation in $\mathbb{E}^d$ of all edges incident to its current location. The robot is constrained to move on edges and vertices. In the 2-dimensional problem one might think of the vertices as locations on a map and the edges as routes connecting these locations. For geometric trees Kleinberg provides a strategy with a competitive ratio of $O(n^{2/3})$, where n is the number of vertices of degree greater than two, and for rectangle packings a strategy with a competitive ratio of $O(n\sqrt{\frac{\log\log n}{\log n}})$, where n is the number of rectangles.

Kleinberg also provides a lower bound of $\Omega(\sqrt{n})$ for the localization problem in geometric trees, which is illustrated by the example in Fig. 1. If the true placement s of the robot is at the bottom of the dth spike to the right of spike t, where $\sqrt{n} < d \leq 2\sqrt{n}$, then a localization strategy has to either explore all spikes between t and s or travel to one of the end points q_1 or q_2 of the base b of the tree. In each case the total distance traveled by the robot is $\Omega(n)$ while the shortest path to localize the robot is the path from s to t, which is of length at most $3\sqrt{n} + 1$.

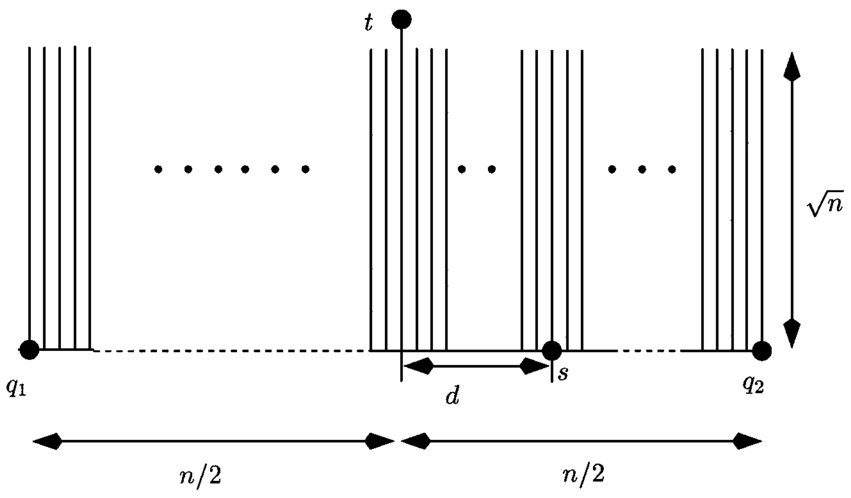

FIG. 1. A geometric tree for which every on-line localization strategy is no better than $\Omega(\sqrt{n})$-competitive.

In this paper we consider only bounded-degree geometric trees as environments, and we present a new localization strategy LPS (Localize-by-Placement-Separation) that has a competitive ratio of $O(\sqrt{n})$. The first and last steps of our strategy are essentially the same as in Kleinberg's $O(n^{2/3})$ strategy. However, in the middle steps we explore periodic patterns in the tree to eliminate all but $O(\sqrt{n})$ possible placements of the robot by traveling no more than $O(\sqrt{n})$ times the length of a shortest path to localize the robot. In view of the example in Fig. 1 our strategy is optimal up to a constant factor and, hence, settles the asymptotic complexity of on-line robot localization in trees, thus solving an open problem posed by Kleinberg.

Another important problem in robotics is searching for a target in an environment [2–6, 10, 14]. If a map of the environment and the position of the target on the map are given, but the robot's position on the map is not given, then Fig. 1 again provides an example of an $\Omega(\sqrt{n})$ lower bound for the competitive ratio of an on-line strategy. We show that our approach can be used to obtain a strategy that solves the searching problem with a competitive ratio of $O(\sqrt{n})$, which is optimal up to a constant factor.

The paper is organized as follows. In the next section we give formal definitions of the main geometric concepts used in the paper, and we give a list of notations. We then present a very rough outline of our new strategy in Section 3 and discuss the first step of the strategy. In Section 4 we show how to identify a critical path in the explored tree that will then be used to guide the further exploration of the robot. The heart of our strategy is contained in Sections 5 and 6, where we show how to expand the subtree known to the robot so that most of the possible placements can be efficiently eliminated. Section 7 then shows that the remaining placements can be eliminated in a greedy fashion. In Section 8 we show how to adapt our strategy to searching for a target in a geometric tree. Finally, in Section 9 we give a short description of an implementation of our algorithm, and we conclude with some open problems in Section 10.

2. DEFINITIONS

In this section we define some of the notation that we use in the paper. The environments that we consider for robot localization are geometric trees.

Definition 2.1. A geometric tree $T = (V, E)$ is a tree embedded into the d-dimensional Euclidean space $\mathbb{E}^d$ such that each $v \in V$ is a point in $\mathbb{E}^d$ and each $e \in E$ is a polygonal path whose end points lie in V. The paths of E intersect only at points in V, and they do not induce any cycles.

We assume that the degree of T is bounded by a constant Δ. We say a vertex of T is *nondegenerate* if it has a degree greater than two. The *size* of T, denoted by $|T|$, is then defined as the number of nondegenerate vertices of T.

It is easy to see that in every (geometric) tree T there is a nondegenerate vertex v_s such that after the removal of v_s each of the remaining subtrees has size at most $|T|/2$. The vertex v_s is called the *split vertex* of T.

As in Kleinberg's work, we assume that the robot knows its current orientation, is able to measure the distance that it has traveled, and has no use of vision other than to know the orientation of all edges incident to its current location. Since the closest nondegenerate vertex v can be reached by performing a two-way spiral search that travels at most nine times the distance from the robot's original location to v [1], we assume that the robot's initial "wake-up" position $\hat{p}$ is located at a nondegenerate vertex of T. We call the possible wake-up locations of the robot *placements*. We denote the set of placements by P. P is not a static set, the more the robot learns about its environment the fewer vertices can be placements. In the beginning P equals the set of all vertices whose incident edges have the same orientation as $\hat{p}$. It is the robot's task to determine $\hat{p}$ by traveling from its wake-up position in T and collecting enough information to rule out all other placements.

In order to describe the motion of the robot we assume that the robot has a local coordinate system that is relative to the robot's wake-up position; that is, the origin $\mathbf{0}$ of the local coordinate system corresponds to $\hat{p}$ in the global coordinate system. We use standard vector addition and scalar multiplication to denote translations and scalings; that is, if α and β are reals, v is a vector and S is a d-dimensional set, then

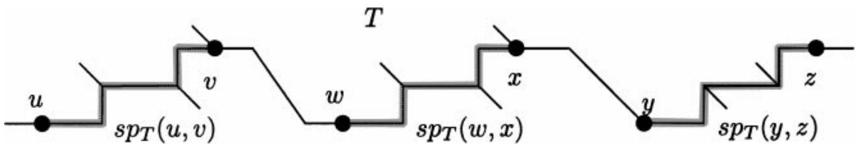

FIG. 2. Path $sp_T(u, v)$ is isomorphic to $sp_T(w, x)$ but not to $sp_T(y, z)$.

$\alpha v + \beta S = \{\alpha v + \beta s \,|\, s \in S\}$. In particular, $v + \mathcal{P}$ is the path $\mathcal{P}$ translated by v, i.e., the path $\mathcal{P}$ starting at node v.

2.1. Paths

We use the standard definition of the length of a polygonal path (using the L_2 norm) for the length $\lambda(\mathcal{P})$ of a path $\mathcal{P}$ as the sum of the lengths of its links. We use $sp_T(p, q)$ to denote the unique simple (shortest) path from point p to point q in the tree T. The *distance* between p and q, denoted by $d_T(p, q)$, is defined as the length of $sp_T(p, q)$. Since the robot is often directed to travel along a nonsimple path in T (for example, when it explores a part of T using depth-first search), for a nonsimple path $\mathcal{P}$ with start point s and end point t, we use $\lambda(\mathcal{P})$ to denote the length traveled by the robot and $\lambda_{sp(\mathcal{P})}$ to denote the length of $sp_T(s, t)$. The *concatenation* of two paths $\mathcal{P}_1$ and $\mathcal{P}_2$ is denoted by $\mathcal{P}_1 * \mathcal{P}_2$, where we assume that the path $\mathcal{P}_2$ is translated such that its start point is the end point of $\mathcal{P}_1$.

We say that path $\mathcal{P}_1$ is *isomorphic* to path $\mathcal{P}_2$ if some translation of $\mathcal{P}_1$ is equivalent to $\mathcal{P}_2$ and all edges adjacent to any vertex on $\mathcal{P}_1$ have the same orientation as the edges adjacent to the corresponding vertex on $\mathcal{P}_2$ (see, for example, Fig. 2). We write $\mathcal{P}_1 \equiv \mathcal{P}_2$ in this case. We say that path $\mathcal{P}_1$ *equals* path $\mathcal{P}_2$ if $\mathcal{P}_1 \equiv \mathcal{P}_2$ and $\mathcal{P}_1$ starts and ends at the same points as $\mathcal{P}_2$. We write $\mathcal{P}_1 = \mathcal{P}_2$ in this case.

If $\mathcal{P}$ is a directed path and $k \geq 1$ is an integer, then the *k-repetition of* $\mathcal{P}$, denoted by $\mathcal{P}^k$, is the path formed by concatenating together k copies of $\mathcal{P}$ in succession. If $0 \leq \tau < 1$, then $\mathcal{P}^\tau$ is the subpath of $\mathcal{P}$ beginning at its start point with length equal to $\tau\lambda(\mathcal{P})$. If $\alpha > 1$, $k = \lfloor \alpha \rfloor$, and $\tau = \alpha - k$, then $\mathcal{P}^\alpha = \mathcal{P}^k * \mathcal{P}^\tau$. $\mathcal{P}^{-1}$ denotes the path from the end point of $\mathcal{P}$ to its start point.

Note that even if $\mathcal{P}$ is a simple path, $\mathcal{P}^\alpha$ may not be (see Fig. 3). In this case $\mathcal{P}^\alpha$ induces a *comb-tree*. A tree is called a comb-tree if it is a geometric tree and there are simple paths $\mathcal{Q}$ and $\mathcal{S}$ such that $\mathcal{Q}^k$ is a simple path and the tree is the union of $\mathcal{Q}^k$ (the *base* of the tree) and the sets $k_i(t - s) + \mathcal{S}$, for $1 \leq i \leq m$, where $0 = k_1 < \cdots < k_m = k$ are natural numbers and s is the start point and t the end point of $\mathcal{Q}$ (see Fig. 3). The m paths of the tree isomorphic to $\mathcal{S}$ are called *spikes*.

2.2. Periodic Paths

If $\mathcal{P}$ is a path such that $\mathcal{P} \equiv \mathcal{Q}^\alpha$, for $\alpha > 1$, then $\mathcal{P}$ is called a *periodic path* and $\mathcal{Q}$ is called a *period of* $\mathcal{P}$. The number α is called the *periodicity of* $\mathcal{P}$ *w.r.t.* $\mathcal{Q}$. If α is an integer, then $\mathcal{Q}$ is called an *integral period of* $\mathcal{P}$. If $\mathcal{P}$ is a path in T that starts with a vertex and $\mathcal{Q}$ is a period of $\mathcal{P}$, then $\mathcal{Q}$ starts and ends with a vertex.

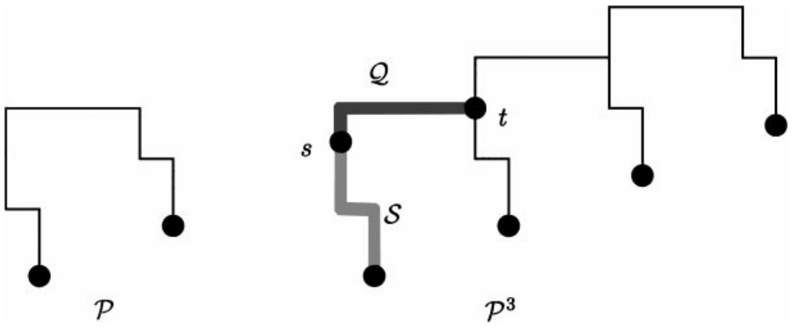

FIG. 3. A path $\mathcal{P}$ such that $\mathcal{P}^3$ is not a simple path. The path $\mathcal{P}^3$ is a comb-tree with base $\mathcal{Q}^3$ and spikes $\mathcal{S}$, $(t - s) + \mathcal{S}$, $2(t - s) + \mathcal{S}$, and $3(t - s) + \mathcal{S}$.

Note that a periodic path may have several different periods. But we will show below that given a set of periods $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_m\}$ of a periodic path $\mathcal{P}$, there is a unique maximal length period $\mathcal{Q}$ of $\mathcal{P}$ that is an integral period of each period $\mathcal{Q}_i$. The period $\mathcal{Q}$ is called the *greatest common period of* $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_m\}$. Its length is the greatest common divisor of the lengths $\lambda(\mathcal{Q}_i)$ (if they are scaled appropriately). The results in this section do not only hold for simple periodic paths, but also for nonsimple periodic paths such as paths that induce comb-trees.

We start by considering two periods of a periodic path. But the lemma can easily be extended to a set of paths $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_m\}$.

LEMMA 2.1. *Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be two paths and let $\mathcal{Q}$ be an integral period of $\mathcal{Q}_1$ and $\mathcal{Q}_2$. If $\mathcal{R}$ is the greatest common period of $\mathcal{Q}_1$ and $\mathcal{Q}_2$, then $\mathcal{Q}$ is an integral period of $\mathcal{R}$.*

Proof. We have $\mathcal{Q}_i \equiv \mathcal{Q}^{k_i} \equiv \mathcal{R}^{l_i}$ for some integers k_i, l_i, where $k_i \geq l_i$ for $i = 1, 2$. Since $\mathcal{R}$ is the longest integral period of $\mathcal{Q}_1$ and $\mathcal{Q}_2$, l_1 and l_2 must be relatively prime. Also, $\mathcal{R} \equiv \mathcal{Q}^{(k_i/l_i)}$, and thus $(k_1/l_1) = (k_2/l_2)$. Since l_1 and l_2 are relatively prime, this equation only holds when k_i is a multiple of l_i, and thus (k_i/l_i) is an integer, which proves the claim. ■

Since finding the greatest common period of two periods of a path is analogous to finding the greatest common divisor of two integers, the proof of the following lemma is similar to Euclid's algorithm for finding the greatest common divisor of two integers.

LEMMA 2.2. *Let $\mathcal{P} \subset T$ be a path that starts with a vertex. If $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are two periods of $\mathcal{P}$ such that $\mathcal{P} \equiv \mathcal{Q}_1^{\alpha_1} \equiv \mathcal{Q}_2^{\alpha_2}$, for some numbers $\alpha_1, \alpha_2 \geq 2$, then there exist two unique nonnegative, relatively prime integers k_1 and k_2 and a period $\mathcal{Q}$ of $\mathcal{P}$ such that $\lambda(\mathcal{Q}_1)/\lambda(\mathcal{Q}_2) = k_1/k_2$ and $\mathcal{Q}_i \equiv \mathcal{Q}^{k_i}$, for $i = 1, 2$. The path $\mathcal{Q}$ is the greatest common period of $\mathcal{Q}_1$ and $\mathcal{Q}_2$.*

Proof. As indicated above, the proof follows the same ideas as used in Euclid's algorithm. Since $\mathcal{P}$ starts at a vertex and has periodicity greater than 1 w.r.t. both $\mathcal{Q}_1$ and $\mathcal{Q}_2$, the paths $\mathcal{Q}_1$ and $\mathcal{Q}_2$ start and end in a vertex. Assume w.l.o.g. that $\alpha_2 \geq \alpha_1$; i.e., $\lambda(\mathcal{Q}_2) \leq \lambda(\mathcal{Q}_1)$. Then, $\mathcal{Q}_2$ is a period of $\mathcal{Q}_1$ since $\mathcal{Q}_1 \equiv \mathcal{Q}_2^{\alpha_2/\alpha_1}$. Let k be the largest integer such that $k\lambda(\mathcal{Q}_2) \leq \lambda(\mathcal{Q}_1)$. Note that if $k\lambda(\mathcal{Q}_2) = \lambda(\mathcal{Q}_1)$ then $\mathcal{Q}_1 \equiv \mathcal{Q}_2^k$; hence, if we let $\mathcal{Q} = \mathcal{Q}_2$, $k_1 = k$, and $k_2 = 1$, then we are done.

If $k\lambda(\mathcal{Q}_2) < \lambda(\mathcal{Q}_1)$, then there is a $0 < \tau < 1$ such that $\mathcal{Q}_1 \equiv \mathcal{Q}_2^k * \mathcal{Q}_2^\tau$. Let $\mathcal{Q}_3 = \mathcal{Q}_2^\tau$. $\mathcal{Q}_3$ is isomorphic to an initial part of $\mathcal{Q}_2$ and a final part of $\mathcal{Q}_1$ (see Fig. 4a). In particular, $\mathcal{Q}_3$ starts and ends with a vertex. And since $\tau < 1$, $\mathcal{Q}_3$ contains fewer vertices than $\mathcal{Q}_2$.

We claim that $\mathcal{Q}_3$ is a period of $\mathcal{Q}_2$. It is easy to prove by induction that $\mathcal{Q}_3^i$ is the initial part of $\mathcal{Q}_2$ for any $i \geq 0$ such that $\lambda(\mathcal{Q}_3^i) \leq \lambda(\mathcal{Q}_2)$. As indicated in Fig. 4b, $\mathcal{Q}_3$ is the initial part of $\mathcal{Q}_2$ and thus also the initial part of $\mathcal{Q}_1$ (because both $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are an initial part of $\mathcal{P}$), so it is repeated right after its first occurrence; i.e., $\mathcal{Q}_3^2$ is the initial part of $\mathcal{Q}_2$, and so on.

If $\mathcal{Q}_3$ is not an integral period of $\mathcal{Q}_2$, then we can repeat the process that we used for $\mathcal{Q}_1$ and $\mathcal{Q}_2$ with $\mathcal{Q}_2$ and $\mathcal{Q}_3$ to find a smaller period of $\mathcal{Q}_3$, and we can continue until we eventually find a path $\mathcal{Q}_k$ that is an integral period of $\mathcal{Q}_{k-1}$ and $\mathcal{Q}_{k-2}$, for some $k \geq 2$. We know that the process halts because $\mathcal{Q}_i$ always contains fewer vertices than $\mathcal{Q}_{i-1}$, for $i \geq 2$. Since $\mathcal{Q}_i = \mathcal{Q}_{i+1}^{\ell_i} * \mathcal{Q}_{i+2}$, for $1 \leq i \leq k-2$ and some integer $\ell_i \geq 1$, $\mathcal{Q}_k$ is also an integral period of $\mathcal{Q}_1$ and $\mathcal{Q}_2$.

Since we have found some integral period of $\mathcal{Q}_1$ and $\mathcal{Q}_2$, there must also exist a longest integral period $\mathcal{Q}$, i.e., the greatest common period. By Lemma 2.1, $\mathcal{Q}$ must be a multiple of $\mathcal{Q}_k$.

Since $\mathcal{Q}$ is the greatest common period of $\mathcal{Q}_1$ and $\mathcal{Q}_2$, $\mathcal{Q}_k^{k_i} \equiv \mathcal{Q}_i$, $i = 1, 2$, for two nonnegative, relatively prime integers k_1 and k_2. Also, $\lambda(\mathcal{Q}_1)/\lambda(\mathcal{Q}_2) = k_1\lambda(\mathcal{Q}_k)/k_2\lambda(\mathcal{Q}_k) = k_1/k_2$. ■

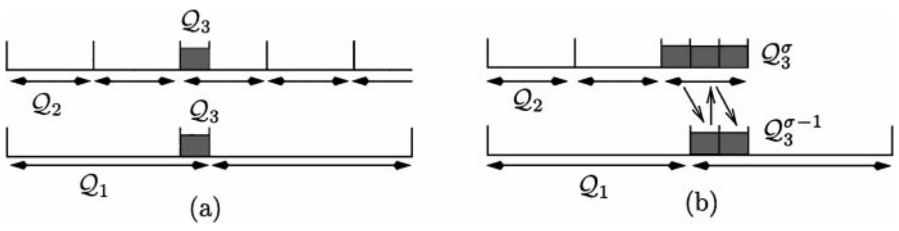

FIG. 4. $\mathcal{Q}_3$ is a period of $\mathcal{Q}_2$ and $\mathcal{Q}_1$.

In the proof above, it is actually not difficult to see that $\mathcal{Q} = \mathcal{Q}_k$. Let $\mathcal{G} = \text{gcp}(\mathcal{Q}_1, \mathcal{Q}_2)$ be the greatest common integral period of $\mathcal{Q}_1$ and $\mathcal{Q}_2$. Assume that $\mathcal{Q}_i \equiv \mathcal{G}^{k_i}$ for some integers k_i, $i = 1, 2$. To show that $\mathcal{Q}_k = \mathcal{G}$, we only need to show that the recursive equation $\text{gcp}(\mathcal{Q}_1, \mathcal{Q}_2) = \text{gcp}(\mathcal{Q}_2, \mathcal{Q}_3)$ holds. We first show that $\mathcal{G}$ is an integral period of $\text{gcp}(\mathcal{Q}_2, \mathcal{Q}_3)$. Since $\mathcal{G}^{k_1} \equiv \mathcal{Q}_1 \equiv (\mathcal{G}^{k_2})^k * \mathcal{Q}_3$, $\mathcal{Q}_3 \equiv \mathcal{G}^{k_1 - kk_2}$, $\mathcal{G}$ is an integral period of $\mathcal{Q}_3$, and by Lemma 2.1, $\mathcal{G}$ is also an integral period of $\text{gcp}(\mathcal{Q}_2, \mathcal{Q}_3)$. Similarly we can show that $\text{gcp}(\mathcal{Q}_2, \mathcal{Q}_3)$ is an integral period of $\mathcal{G}$, so $\mathcal{G} = \text{gcp}(\mathcal{Q}_2, \mathcal{Q}_3)$ and therefore $\mathcal{Q}_k = \mathcal{G}$.

Note that in the above lemma, $\lambda(\mathcal{Q}) = \gcd(\lambda(\mathcal{Q}_1), \lambda(\mathcal{Q}_2))$ if $\lambda(\mathcal{Q}_1)$ and $\lambda(\mathcal{Q}_2)$ are scaled to be integers. Observe also that it is necessary for $\mathcal{P}$ to contain vertices in order for the lemma to be true. Suppose, for example, that $\mathcal{P}$ is a straight line segment of length 2π containing no vertices. Then a line segment $\mathcal{Q}_1$ of length π and a line segment $\mathcal{Q}_2$ of length 2 would both be periods of $\mathcal{P}$, but there is no path that is an integral period of both $\mathcal{Q}_1$ and $\mathcal{Q}_2$. Lemma 2.2 can also be extended to a set of periods of a path $\mathcal{P}$.

LEMMA 2.3. *If $\mathcal{P}$ is a periodic path and $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_m\}$ is a set of periods of $\mathcal{P}$ such that the periodicity of $\mathcal{P}$ w.r.t. each $\mathcal{Q}_i$ is greater than or equal to two, then there exists a unique period $\mathcal{Q}$ of $\mathcal{P}$, called the greatest common period of $\mathcal{P}$ w.r.t. $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_m\}$, such that*

(i) *$\mathcal{Q}$ is an integral period of $\mathcal{Q}_i$, for all $1 \le i \le m$, and*

(ii) *if $\mathcal{R}$ is an integral period of $\mathcal{Q}_i$, for all $1 \le i \le m$, then $\mathcal{R}$ is also an integral period of $\mathcal{Q}$.*

Proof. The proof is by induction on m. For $m = 1$ the claim is trivial. Assume that it is true for all sets containing $m - 1$ periods, and let $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_m\}$ be a set of m periods. By the induction hypothesis there is a period $\mathcal{Q}'$ of $\mathcal{P}$ with the desired properties w.r.t. $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_{m-1}\}$. By Lemma 2.2 there is a greatest common period $\mathcal{Q}$ of $\mathcal{Q}_m$ and $\mathcal{Q}'$. Since $\mathcal{Q}$ is the longest integral period of $\mathcal{Q}'$ and $\mathcal{Q}_m$, it is also the longest integral period of all $\mathcal{Q}_i$, $1 \le i \le m$, and thus part (ii) holds by Lemma 2.1. ∎

COROLLARY 2.1. *If $\mathcal{P}$ is a periodic path and $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_m\}$ is the set of all periods of $\mathcal{P}$ such that the periodicity of $\mathcal{P}$ w.r.t. each $\mathcal{Q}_i$ is greater than or equal to two, then there exists a unique longest period $\mathcal{Q}_j$ of $\mathcal{P}$ such that $\mathcal{Q}_j$ is an integral period of $\mathcal{Q}_i$, for all $1 \le i \le m$.*

2.3. Notation

The following is a summary of some of the notation we use in the paper. Note that the last four items are introduced in the next section.

- T—the geometric tree where the robot is located.
- n—the number of nondegenerate vertices of T.
- $sp_T(u, v)$—the unique shortest path between u and v in T.
- $\lambda(\mathcal{P})$; $\lambda_{sp(\mathcal{P})}$—the length of path $\mathcal{P}$; the length of the shortest path from the start to the end point of $\mathcal{P}$.
- $\hat{p}$—the robot's wake-up position in T.
- P—the set of possible placements of the robot in T (shrinking during the localization).
- $\hat{T}$—the geometric tree T in the robot's local coordinate system ($= -\hat{p} + T$).
- $\mathbf{0}$—the origin of $\hat{T}$ (corresponds to $\hat{p}$ in T).
- $\hat{T}_{ex}$—the part of $\hat{T}$ explored by the robot in the robot's local coordinate system.
- $\hat{T}_{ex}^{S1}$—the part of $\hat{T}$ explored by the robot in Step 1.
- ϱ—a lower bound on the length of a shortest path to localize the robot found in Step 1.
- U—the set of p-vertices in $\hat{T}_{ex}$ (see Section 4.1).

3. A NEW STRATEGY FOR ROBOT LOCALIZATION

Our localization strategy consists of five main steps. The following pseudo-code gives a rough impression of how the strategy works. It mainly serves as a guide through the subsequent sections where each step will be explained in detail.

```
Strategy Localize-by-Placement-Separation (LPS)
Input:     A geometric tree $T = (V, E)$ with $n$ nondegenerate vertices and a wake-up position of
           the robot;
Output:    The position of the robot in $T$;

Variables: The set of placements $P$;
           The tree $\hat{T}_{ex}$ that has been explored;

$P \leftarrow$ set of vertices $v \in V$ whose incident edges have the same orientation as $\hat{p}$;
$\hat{T}_{ex} \leftarrow \{\mathbf{0}\}$;
Step 1: Perform a spiral search until either $|\hat{T}_{ex}| = \sqrt{n}$ or $|P| \leq 2\sqrt{n}$;
        Translate the origin to the split vertex of $\hat{T}_{ex}$;
        $\hat{T}_{ex}^{S1} \leftarrow \hat{T}_{ex}$;
if $|P| > 2\sqrt{n}$ then
   Step 2: $P_{triv} \leftarrow$ the set of all sparse placements and placements inducing a comb-tree;
           $P = P - P_{triv}$;
           Identify a path $\mathcal{C}_{ex}$ in $\hat{T}_{ex}^{S1}$ containing all other placements;
           $\mathcal{R}_{cr} \leftarrow \{(p, p') \in P \times P \mid p' \in p + \hat{T}_{ex}^{S1}\}$;
   if $\mathcal{R}_{cr} \neq \emptyset$ then
      Step 3: Extend $\mathcal{C}_{ex}$ to a periodic path $\mathcal{P}_{ex}$ with at least $\sqrt{n}$ vertices;
   if $\mathcal{R}_{cr} \neq \emptyset$ then
      Step 4: Extend $\mathcal{P}_{ex}$ until $\mathcal{R}_{cr} = \emptyset$;
Step 5: Eliminate the remaining placements and the placements in $P_{triv}$ using a greedy strategy;
end Localize-by-Placement-Separation
```

It should be noted that in the above strategy Steps 1 and 5 are essentially the same steps as in the strategy proposed by Kleinberg [11]. The only exception is that in Kleinberg's strategy up to $n^{2/3}$ vertices are visited by the spiral search in Step 1. After analyzing each step of the strategy we will be able to prove the following theorem.

THEOREM 3.1. *Let T be a geometric tree. Strategy* LPS *achieves a competitive ratio of $O(\sqrt{n})$ for localizing a robot in T.*

Before we give the detailed description of each step we give a brief overview of the strategy.

Our goal is to use a simple greedy strategy (Step 5 of our strategy, see Section 7). For each pair of placements there is a shortest distinguishing path. So we could just explore these paths in order of increasing length until we have identified the wake-up position. Unfortunately, this strategy is only k-competitive, where k is the number of placements. So we can only use it if there are at most $O(\sqrt{n})$ placements. Otherwise, we must first reduce the number of placements before we can run the greedy strategy.

Our algorithm runs in five steps. The pseudo-code was given above. Of course, the robot would stop at any time when it has identified its wake-up position. In the first step of the algorithm, the spiral search (see Section 3.1), the robot explores its nearby environment by successive depth-first searches, where the search depth is doubled after each iteration. One could imagine the robot's wake-up position as the center of concentric circles. The robot stops this exploration when it has visited at least $\sqrt{n}$ different vertices during the last iteration.

Then the robot moves to the center of the explored tree, the *split-vertex*, and assumes that this was the wake-up position. This allows us to argue that either there are not many placements left (Corollary 3.1) or all placements lie on the same periodic path (Lemma 4.3). From the split vertex at most $\frac{\sqrt{n}}{2}$ vertices in the explored tree can be reached by traveling in any direction. If there are at most $O(\sqrt{n})$ placements (for example, if no translation of the explored tree contains another placement, see Corollary 3.1) then the robot switches to the greedy strategy whose competitive ratio is linear in the number of placements.

Otherwise, after the spiral search the robot restricts its search to one long path, the periodic path (see Section 4.1), which depends on what it learned in the first step. It turns out that there is one periodic path, the *critical path*, containing most of the placements in the explored tree (see Section 4.2). Eliminating these placements in Steps 3 and 4 leaves at most $O(\sqrt{n})$ placements (Lemma 4.3 and Corollary 3.1) so that we can continue with the greedy strategy.

In Step 3 (see Section 5) the robot explores branches emanating from the critical path. The path plus branches is called the *periodic path*. The robot stops either if the periodic path has $\sqrt{n}$ vertices or if there are no more critical pairs. A pair of placements (p, p') is a *critical pair* if $p' \in p + \hat{T}_{ex}$. In that case the robot continues with the greedy strategy.

In Step 4 (see Section 6) the robot extends the periodic path in the following way. Since the path has at least $\sqrt{n}$ nodes, the periodic tree induced by the $2\sqrt{n}$-repetition of the periodic path has at least $2n$ nodes. Therefore there are points called *mismatches* where the periodic tree and the tree T differ locally. The paths to these mismatches are called *mismatch paths*. It turns out that a certain collection of only $\log n$ mismatch paths contains at least one mismatch path for each placement (Corollary 6.1). The robot explores these paths in order of increasing length until it finds an initial mismatch (see Section 6.1). After finding this mismatch, the robot tries to find the shortest mismatch path. This is called Mismatch-Propagation and it is the most complicated part of the strategy. It will be described in Section 6.2. After the robot has found the path to the nearest mismatch there are no critical pairs left. After this the robot starts the final greedy strategy (see Section 7).

3.1. Step 1: Spiral Search

In Step 1 of Strategy LPS the robot performs a spiral search [1] as follows. Starting at the origin it performs successive depth-first searches, each time visiting all points within distance d of the origin (where initially d is the length of the shortest edge of T) and then doubling d if it has not yet localized itself. During the search the robot keeps track of the size $|\hat{T}_{ex}|$ of the explored subtree $\hat{T}_{ex}$ and the set P of all possible placements of the robot, i.e., the set of all $p \in T$ such that $p + \hat{T}_{ex} \subseteq T$. It continues searching until $|\hat{T}_{ex}| = \sqrt{n}$ or $|P| \leq 2\sqrt{n}$. If $|P| > 2\sqrt{n}$, then the robot continues with Step 2; otherwise, it skips Steps 2–4 and only executes Step 5 (the greedy strategy).

If d is the current search depth, then the ball of radius $\varrho = d/2$ around $\hat{p}$ has been completely explored. Clearly, ϱ is a lower bound on the distance that the robot needs to travel in order to localize. The distance traveled during the spiral search is at most $8\Delta\sqrt{n}d = 16\Delta\sqrt{n}\varrho = O(\sqrt{n}\varrho)$ since the maximum degree of each vertex is at most Δ [11]. We will show that the remaining steps of the algorithm can also be implemented such that the distance traveled is at most $O(\sqrt{n}\varrho)$.

After the spiral search the robot moves to the split vertex v_s of $\hat{T}_{ex}$ and considers this vertex from now on to be the origin, that is, all points v in the local coordinate system of the robot are translated to $-v_s + v$. This allows us to argue that either there are not many placements (Corollary 3.1) or all placements lie on the same periodic path (Lemma 4.3). For simplicity, we also refer to the translated explored tree as $\hat{T}_{ex}$. Note that the translation increases the maximum distance from the new origin to the farthest leaf of $\hat{T}_{ex}$ by at most d, so the maximum distance from the origin to a leaf of $\hat{T}_{ex}$ is now $2d = 4\varrho$ (which is also an upper bound on the diameter of $\hat{T}_{ex}$). This fact will be used in Lemma 4.5.

LEMMA 3.1. *If $p + \hat{T}_{ex}$ contains no placements different from p, for all $p \in P$, then*

$$\left|\bigcup_{p\in P}(p + \hat{T}_{ex})\right| \geq \frac{k+l}{2} \cdot |\hat{T}_{ex}|,$$

where $k = |P|$ and l is the number of connected components of $\bigcup_{p\in P}(p + \hat{T}_{ex})$.

Proof. We prove by induction on k: If P' is a set of k placements of some tree T_0 (where the origin of T_0 is the split vertex of T_0) such that $p + T_0$ contains no other placements from P' than p, for all $p \in P'$, then

$$|T_{P'}| \geq \frac{k+l}{2} \cdot |T_0|,$$

where $T_{P'} = \bigcup_{p\in P'}(p + T_0)$ and l is the number of connected components of $T_{P'}$.

The claim clearly holds if $k = 1$. If $k > 1$ then let $P' = \{p_1, \ldots, p_k\}$. For $i = 1, \ldots, k$ let $T_i = \bigcup_{p\in P'-\{p_i\}}(p + T_0)$. Note that $p_i \notin T_i$.

If there is an i such that $(p_i + T_0) \cap T_i = \emptyset$ then $p_i + T_0$ is one of the l connected components of $T_{P'}$ and thus

$$|T_{P'}| = |T_i| + |p_i + T_0| \geq \frac{(k-1)+(l-1)}{2} \cdot |T_0| + |T_0| = \frac{k+l}{2} \cdot |T_0|.$$

Otherwise, each $p_i + T_0$ overlaps with at least one other $p_j + T_0$. Any $p_i + T_0$ is split into $deg(p_i)$ many components (subtrees) if p_i is removed. Since T is acyclic there must be one i such that only one of these $deg(p_i)$ components overlaps with T_i. But that component can have size at most $|T_0|/2$. Since T_i still has l components we have

$$|T_{P'}| \geq |T_i| + \frac{|T_0|}{2} \geq \frac{(k-1)+l}{2} \cdot |T_0| + \frac{|T_0|}{2} = \frac{k+l}{2} \cdot |T_0|. \qquad \blacksquare$$

COROLLARY 3.1. *Assume $p + \hat{T}_{ex}$ contains no placements different from p, for all $p \in P$. If $|\hat{T}_{ex}| \geq \sqrt{n}$ then $|P| \leq 2\sqrt{n}$.*

Proof. Lemma 3.1 implies

$$n = |T| \geq \frac{|P|}{2} \cdot |\hat{T}_{ex}| \geq \frac{\sqrt{n}}{2} \cdot |P|$$

and, therefore, $|P| \leq 2\sqrt{n}$. $\blacksquare$

This suggests we should try to eliminate placements that are contained in $p + \hat{T}_{ex}$ for some other placement p. After we have done this we can continue with the greedy strategy.

4. STEP 2: IDENTIFYING A CRITICAL PATH

Let $\hat{T}_{ex}^{S1}$ denote the tree that has been explored in Step 1. If after Step 1 there are at least two placements p_1 and p_2 with $p_2 \in p_1 + \hat{T}_{ex}^{S1}$, then in Step 2 the robot identifies a critical path of $\hat{T}_{ex}^{S1}$, and in Steps 3 and 4 it travels along an extension of this path eliminating placements until there are no pairs p_1, p_2 left with $p_2 \in p_1 + \hat{T}_{ex}^{S1}$. We note that the robot does not need to travel to identify the critical path (but it must travel to explore it). In Steps 3 and 4 it will travel a distance of at most $O(\sqrt{n}_{\mathcal{Q}})$. After Step 4 at most $2\sqrt{n}$ placements remain and the robot uses the greedy strategy to eliminate all placements but one.

4.1. The Critical Path of a Placement

We say a vertex $v \in \hat{T}_{ex}^{S1}$ is a *p-vertex* if there is a placement q such that the node $v + q$ in T is also a placement. We denote the set of all p-vertices in $\hat{T}_{ex}^{S1}$ by U. We call a placement p *sparse* if $p + \hat{T}_{ex}^{S1}$ contains at most three placements (including p), and *dense* otherwise.

If $U' = \{v_1, \ldots, v_k\}$ is a set of $k \geq 2$ vertices of T then U' *induces a periodic path* if there is a path $\mathcal{Q}$ such that, for all $1 \leq i \leq k-1$, either $sp_T(v_i, v_{i+1}) \equiv \mathcal{Q}^{l_i}$ or $sp_T(v_{i+1}, v_i) \equiv \mathcal{Q}^{l_i}$, for some natural numbers l_i.

Kleinberg proved a crucial lemma stating that all placements that are contained in $p + \hat{T}_{ex}^{S1}$ induce either a simple periodic path or a comb-tree in $p + \hat{T}_{ex}^{S1}$ (and thus also in T) [11]. Note that a simple periodic path is a degenerate comb-tree.

LEMMA 4.1 [11]. *If p is a placement in P then either the set of all placements that are contained in $p + \hat{T}_{ex}^{S1}$ is equal to $\{p\}$ or it induces a comb-tree in T.*

If the placements in $p + \hat{T}_{ex}^{S1}$ induce a simple periodic path in T then we call it the *critical path $\mathcal{C}_p$ of p*. This path corresponds to a path $\hat{\mathcal{C}}_p = -p + \mathcal{C}_p$ in $\hat{T}_{ex}^{S1}$.

Let P_{triv} be the set of all placements contained in all the $p + \hat{T}_{ex}^{S1}$, where p is either a sparse placement or a placement on the spike of a nondegenerate comb-tree.

LEMMA 4.2. $|P_{triv}| = O(\sqrt{n})$.

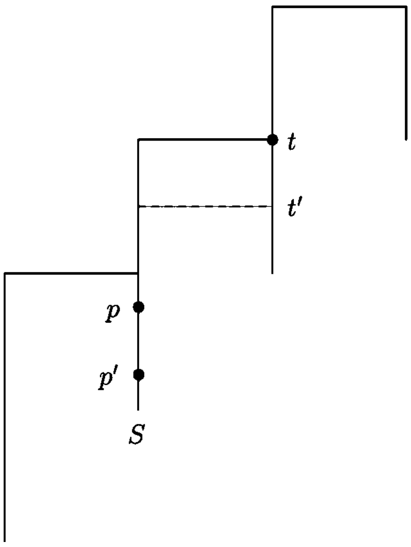

FIG. 5. If p and p' were placements on spike S of a comb-tree in $\hat{T}_{ex}^{S1}$ then the translation of the path from p to t starting at p' and ending in t' on the next spike would also exist, creating a cycle.

Proof. Analogously to Corollary 3.1 we see that at most $6\sqrt{n}$ placements can be contained in all the $p + \hat{T}_{ex}^{S1}$, where p is a sparse placement.

If p is a placement inducing a nondegenerate comb-tree in $\hat{T}_{ex}^{S1}$ then p lies on a spike S of this comb-tree. But then no other placement can lie on S (in $\hat{T}_{ex}^{S1}$); otherwise there would be a cycle between S and the next spike of the comb-tree (see Fig. 5). Thus, p lies at the end of a spike. Since the origin of $\hat{T}_{ex}^{S1}$ is its split vertex, removing p would cut off at least $\frac{\sqrt{n}}{2}$ vertices from $\hat{T}_{ex}^{S1}$ which are not p-vertices. Therefore there can be at most $2\sqrt{n}$ placements on spikes of nondegenerate comb-trees. ■

We will from now on ignore the placements in P_{triv} until we reach the greedy strategy. For simplicity, we also refer to the set of the remaining placements (dense and not on a comb-tree) as P. If this new set P contains at most $O(\sqrt{n})$ placements we immediately proceed with the greedy strategy. Otherwise, we must try to eliminate many placements in P.

4.2. Computing a Unique Critical Path

For sparse placements p and p' it can happen that $\hat{\mathcal{C}}_p$ and $\hat{\mathcal{C}}_{p'}$ do not lie on the same periodic path (see Fig. 6). But this cannot happen for dense placements.

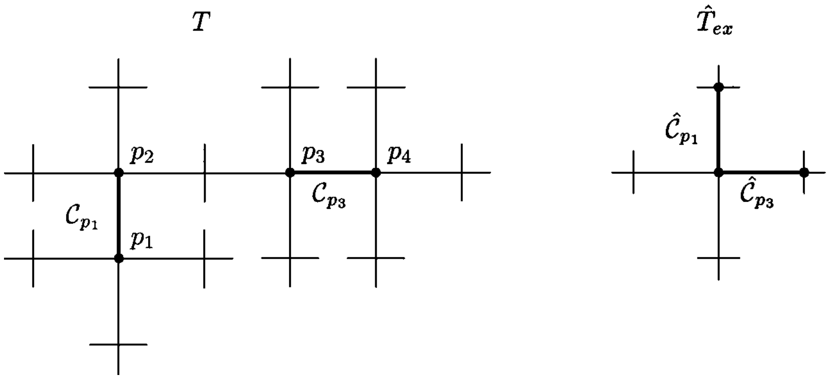

FIG. 6. The points p_1, p_2, p_3, and p_4 are sparse placements of $\hat{T}_{ex}$ in T, and the critical path $\hat{\mathcal{C}}_{p_1}$ of p_1 does not belong to the same periodic path as the critical path $\hat{\mathcal{C}}_{p_3}$ of p_3.

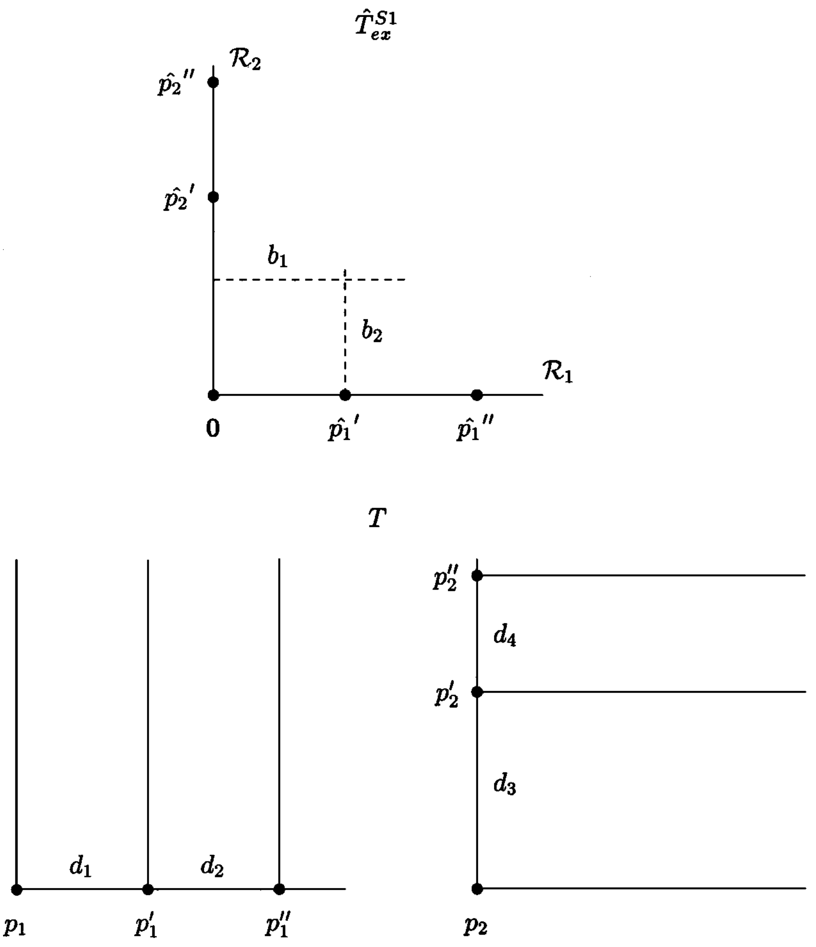

FIG. 7. Two dense placements p_1 and p_2 induce two different periodic paths $\hat{\mathcal{R}}_1$ and $\hat{\mathcal{R}}_2$ in $\hat{T}^{S1}_{ex}$. In this case, $d_i' = d_i$ for $i = 1, \dots, 4$, $b_1 = d_3$, and $b_2 = d_1 = d_2$, so $\hat{T}^{S1}_{ex}$ also contains the two dotted paths. Since $b_1 > d_1$, placing $\hat{T}^{S1}_{ex}$ on p_1 would create a cycle (not to mention that the dotted paths already form a cycle in $\hat{T}^{S1}_{ex}$).

LEMMA 4.3. *If p and p' are dense placements then $\hat{\mathcal{C}}_p$ and $\hat{\mathcal{C}}_{p'}$ are part of the same simple periodic path in $\hat{T}^{S1}_{ex}$.*

Proof. The proof is by contradiction. Let p_1 and p_2 be two dense placements that induce two different periodic paths $\hat{\mathcal{R}}_1$ and $\hat{\mathcal{R}}_2$ in $\hat{T}^{S1}_{ex}$ (see Fig. 7). Since $p_1 + \hat{T}^{S1}_{ex}$ (and $p_2 + \hat{T}^{S1}_{ex}$) contain at least four placements we can assume there are two placements p_1' and p_1'' to the right of p_1 in $p_1 + \hat{T}^{S1}_{ex}$ and two placements p_2' and p_2'' above p_2 in $p_2 + \hat{T}^{S1}_{ex}$ (we use "right" and "above" referring to Fig. 7, but of course the picture could also be flipped or rotated).

Let $d_1 = d(p_1, p_1')$, $d_2 = d(p_1', p_1'')$, $d_3 = d(p_2, p_2')$, and $d_4 = d(p_2', p_2'')$. Since p_1 and p_1' are placements, the paths $p_1 + \hat{\mathcal{R}}_2$ and $p_1' + \hat{\mathcal{R}}_2$ exist in T. Thus, the exploration if started in p_1 would have seen some initial part of $p_1' + \hat{\mathcal{R}}_2$. If the current origin $\mathbf{0}$ was the original wake-up position where we started the spiral search in Step 1 (i.e., explored the tree up to the same distance in all directions) then we would have seen a subpath of length d_2 of $\hat{\mathcal{R}}_2$ starting at p_1' (or the full path, if $d_3 + d_4 < d_2$). Unfortunately, we moved the origin to the split vertex of $\hat{T}_{ex}$ after the spiral search. But wherever the origin was originally, we must have seen a subpath of $\hat{\mathcal{R}}_2$ starting at p_1' of length $d_2' = \min\{d_2, d_3 + d_4\}$ (either the original origin was to the right of the current origin, then we have seen at least a subpath of length $d_3 + d_4$, or else we have seen at least a subpath of length d_2).

Similarly, placing $\hat{T}^{S1}_{ex}$ on p_1' we must also have seen a subpath of $\hat{\mathcal{R}}_2$ of length $d_1' = \min\{d_1, d_3 + d_4\}$, starting at p_1''.

Analogously, let $d_3' = \min\{d_3, d_1 + d_2\}$ and $d_4' = \min\{d_4, d_1 + d_2\}$. Let $b_1 = \max\{d_3', d_4'\}$ and $b_2 = \max\{d_1', d_2'\}$. Then $\hat{T}^{S1}_{ex}$ contains a subpath of $\hat{\mathcal{R}}_1$ of length b_1 starting at distance $\min\{d_3, d_4\}$ from $\mathbf{0}$ on $\hat{\mathcal{R}}_2$ and a subpath of $\hat{\mathcal{R}}_2$ of length b_2 starting at distance $\min\{d_1, d_2\}$ from $\mathbf{0}$ on $\hat{\mathcal{R}}_1$.

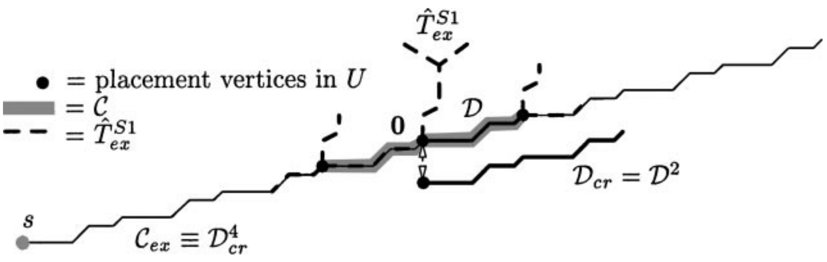
$\hat{T}_{ex}^{S1}$
● = placement vertices in U
= $\mathcal{C}$
= $\hat{T}_{ex}^{S1}$
$\mathbf{0}$
$\mathcal{D}$
$\mathcal{D}_{cr} = \mathcal{D}^2$
s
$\mathcal{C}_{ex} \equiv \mathcal{D}_{cr}^4$

FIG. 8. The explored tree $\hat{T}_{ex}^{S1}$ and $\mathcal{D}_{cr}$. $\mathcal{C}_{ex} = \mathcal{D}_{cr}^4$ starts at s.

In Fig. 7, these two new subpaths already form a cycle in $\hat{T}_{ex}^{S1}$, which is impossible. But that is not always the case. However, it must be the case that either $b_1 \geq d_1$ or $b_2 \geq d_3$. In the former case, placing $\hat{T}_{ex}^{S1}$ on p_1 creates a cycle, and in the latter case, placing $\hat{T}_{ex}^{S1}$ on p_2 creates a cycle. But that is impossible. ■

We denote the simple periodic path in $\hat{T}_{ex}^{S1}$ containing all p-vertices by $\mathcal{C}$. Since $\mathcal{C}$ is periodic and all p-vertices $v \in U$ correspond to dense placements, by Corollary 2.1 there is a unique integral period $\mathcal{D}$ of $\mathcal{C}$; i.e., for all $v \in U$ there is an integer k with either $sp_{\hat{T}_{ex}^{S1}}(\mathbf{0}, v) \equiv \mathcal{D}^k$ or $sp_{\hat{T}_{ex}^{S1}}(v, \mathbf{0}) \equiv \mathcal{D}^k$. We assume w.l.o.g. that $\mathcal{D}$ starts at the origin. Let k be the minimum integer such that $\lambda(\mathcal{D}^k) \geq \lambda(\mathcal{C})$. Let $\mathcal{D}_{cr} = \mathcal{D}^k$ where the start point of $\mathcal{D}_{cr}$ is the origin (see Fig. 8). Since the diameter of $\hat{T}_{ex}^{S1}$ is at most 4ϱ and $k \geq 3$, $\lambda(\mathcal{C}) \leq 4\varrho$ and $\lambda(\mathcal{D}_{cr}) \leq \frac{3}{2}\lambda(\mathcal{C}) \leq 6\varrho$.

We call a pair of placements (p, p') a *critical pair* if $p' \in p + \mathcal{D}_{cr}$. The set of all critical pairs is denoted by $\mathcal{R}_{cr}$. A vertex $v \in \mathcal{C}$ in $\hat{T}_{ex}$ is called a *critical vertex* if there is a critical pair (p, p') with $p' - p = v$. In the beginning the critical vertices are just the p-vertices. Our aim is to eliminate all critical vertices because then we can continue with the greedy strategy by Corollary 3.1 and the lemma below.

LEMMA 4.4. *If $\mathcal{R}_{cr} = \emptyset$ then $(p + \hat{T}_{ex}^{S1})$ contains no placements different from p, for all $p \in P$.*

Proof. The proof is by contradiction. Assume that $\mathcal{R}_{cr} = \emptyset$ and there are two placements p and p' with $p' \in p + \hat{T}_{ex}^{S1}$. Let $v = p' - p$ be the p-vertex that corresponds to the pair (p, p'). The path $\mathcal{P}_v$ from the origin to v is the l-repetition $\mathcal{D}^l$ of the period $\mathcal{D}$ of $\mathcal{C}$ for an integer l. Since $\lambda(\mathcal{C}) \geq \lambda(\mathcal{D}^l)$, either $p' \in p + \mathcal{D}_{cr}$ or $p \in p' + \mathcal{D}_{cr}$. Hence, either (p', p) or (p, p') is a critical pair—a contradiction. ■

Since, for a critical pair $(p, p') \in \mathcal{R}_{cr}$, by definition $p' \in p + \mathcal{D}_{cr}$, the following observation is immediate.

Observation 4.1. If $(p, p') \in \mathcal{R}_{cr}$, then $d_T(p, p') \leq \lambda(\mathcal{D}_{cr})$.

The robot now explores the path $\mathcal{D}_{cr}^2$ from the origin $\mathbf{0}$ of $\hat{T}_{ex}^{S1}$, returns to $\mathbf{0}$, and then explores $\mathcal{D}_{cr}^{-2}$. Note that this may remove some placements from P. In particular, if one of the two paths does not exist all critical pairs will be eliminated and we can proceed with the greedy strategy. We make $s = \mathbf{0} + \mathcal{D}_{cr}^{-2}$ the new origin $\mathbf{0}$ of $\hat{T}_{ex}^{S1}$ and accordingly translate all points v in the local coordinate system of the robot to $-s + v$. Note that a critical pair before the translation remains a critical pair after the translation.

We call $\mathcal{D}_{cr}^4$ the *critical path* $\mathcal{C}_{ex}$ *of* $\hat{T}_{ex}^{S1}$ (see Fig. 8). The length of $\mathcal{C}_{ex}$ is between $4\lambda(\mathcal{C})$ and $6\lambda(\mathcal{C}) \leq 24\varrho$. Since $\mathcal{C}$ is contained in $\hat{T}_{ex}^{S1}$, the number of vertices in $\mathcal{C}$—and, thus, in $\mathcal{C}_{ex}$—is at most $O(\sqrt{n})$. We summarize the properties of $\mathcal{C}_{ex}$ in the following lemma.

LEMMA 4.5. *The critical path $\mathcal{C}_{ex}$ of $\hat{T}_{ex}^{S1}$ has length at least $4\lambda(\mathcal{C})$ and at most 24ϱ. It contains at most $O(\sqrt{n})$ vertices.*

The following observation follows directly from the fact that $\mathcal{D}^{4k} \equiv \mathcal{D}_{cr}^4 \equiv \mathcal{C}_{ex}$ and that, for each p-vertex v, there exists an integer $k' \leq k$ such that either $sp_{\hat{T}_{ex}^{S1}}(\mathbf{0}, v) \equiv \mathcal{D}^{k'}$ or $sp_{\hat{T}_{ex}^{S1}}(v, \mathbf{0}) \equiv \mathcal{D}^{k'}$.

Observation 4.2. If $(p, p') \in \mathcal{R}_{cr}$ and $v = p' - p$, then $sp_{\hat{T}_{ex}^{S1}}(\mathbf{0}, v)$ is a period of $\mathcal{C}_{ex}$.

4.3. The Critical Period

Recall that by Observation 4.2 if $(p, p') \in \mathcal{R}_{cr}$, then $sp_{\hat{T}_{ex}^{S1}}(\mathbf{0}, p' - p)$ is a period of $\mathcal{C}_{ex}$ with $d_T(p, p') \le \lambda(\mathcal{D}_{cr}) \le \lambda(\mathcal{C}_{ex})/4$. This gives rise to the following definition.

DEFINITION 4.1. The greatest common period of the set

$$\{sp_{\hat{T}}(\mathbf{0}, p' - p) \mid (p, p') \in \mathcal{R}_{cr}\}$$

is called the *critical period of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{R}_{cr}$* and denoted by $\mathcal{D}_{ex}$.

Note that the critical period depends only on $\mathcal{R}_{cr}$. The following simple observation follows directly from the definition of the critical period.

Observation 4.3. $d_{\hat{T}}(\mathbf{0}, p' - p) \ge \lambda(\mathcal{D}_{ex})$, for all $(p, p') \in \mathcal{R}_{cr}$.

The following observation is based on the properties of periodic paths.

LEMMA 4.6. *Let $\mathcal{Q}_{cr}$ be a subset of $\mathcal{R}_{cr}$. If $\mathcal{D}_{ex}$ is the critical period of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{R}_{cr}$ and $\mathcal{E}_{ex}$ is the critical period of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{Q}_{cr}$, then $\mathcal{D}_{ex}$ is an integral period of $\mathcal{E}_{ex}$. That is, there is an integer $k \ge 1$ such that $\mathcal{E}_{ex} = \mathcal{D}_{ex}^k$.*

Proof. Since $\mathcal{Q}_{cr} \subseteq \mathcal{R}_{cr}$, $\mathcal{D}_{ex}$ is an integral period of each path $sp_{\hat{T}}(\mathbf{0}, p' - p)$ with $(p, p') \in \mathcal{Q}_{cr} \subseteq \mathcal{R}_{cr}$. Thus by Lemma 2.1, $\mathcal{D}_{ex}$ is an integral period of $\mathcal{E}_{ex}$, as claimed. ■

One consequence of Lemma 4.6 is that if $\mathcal{D}_{ex}$ changes, then it at least doubles in length (see Lemma 6.2 and Section 5.2).

5. STEP 3: EXPLORING A PERIODIC PATH

If there are critical pairs of placements left after Step 2 then the robot explores branches emanating from the critical path $\mathcal{C}_{ex}$ in Step 3 until either it has found a nonsimple periodic path $\mathcal{P}$ with at least $\sqrt{n}$ vertices or there are no more critical pairs. If there are no more critical pairs then there are at most $2\sqrt{n}$ placements left (see Corollary 3.1) and the robot continues with the greedy strategy. Otherwise, the robot continues with Step 4 (Mismatch-Propagation).

After the robot has identified the critical path $\mathcal{C}_{ex}$ in Step 2 of Strategy LPS, it considers the $\sqrt{n}$ vertices of $\hat{T}_{ex}^{S1}$ sorted by increasing distance to the (new) origin $\mathbf{0}$ of $\hat{T}_{ex}^{S1}$. For each vertex v, the robot tries to visit its "neighbors" which are located at multiples of $\mathcal{D}_{ex}$, that is, at the end points of the path $v + \mathcal{D}_{ex}^i$, for $1 \le i \le 2k$. For each $1 \le i \le 2k$, it either succeeds in traveling to such a point or it finds the first point at which the path to $v + \mathcal{D}_{ex}^i$ differs from T.

To describe this strategy more precisely, we first define a few terms. Let v be a vertex in T and $\mathcal{P}$ be a path in T. The closest point of $\mathcal{P}$ to v is called the $\mathcal{P}$-*base* of v. The path $\mathcal{S}$ from the $\mathcal{P}$-base v' of v to v is called the *spike of v w.r.t. $\mathcal{P}$*. The path that is given by the part of $\mathcal{P}$ from its start point to v' concatenated with $\mathcal{S} * \mathcal{S}^{-1}$ and the remaining part of $\mathcal{P}$ is called the path $\mathcal{P}$ *augmented by* v (see Fig. 9a).

Let v be a vertex in $\hat{T}_{ex}$ and v' be the $\mathcal{C}_{ex}$-base of v. There is an integer k and a $0 \le \tau < 1$ depending on the critical period $\mathcal{D}_{ex}$ such that v' is the end point of $\mathcal{D}_{ex}^k * \mathcal{D}_{ex}^\tau$. Let $\mathcal{D}_v$ be the concatenation of $\mathcal{D}_{ex}^\tau$ with the spike of v w.r.t. $\mathcal{C}_{ex}$. For an integer i, the last point where the path $\mathcal{D}_{ex}^i * \mathcal{D}_v$ is isomorphic to a path in T starting at $\hat{p}$ is called the *absolute $\mathcal{D}_{ex}^i$-neighbor* of v and is denoted by $v \oplus \mathcal{D}_{ex}^i$. See Fig. 9b for an illustration. The *relative $\mathcal{D}_{ex}^i$-neighbor* is the last point where the path $\mathcal{D}_{ex}^i * \mathcal{D}_{ex}^k * \mathcal{D}_v$ is isomorphic to a path in T starting at $\hat{p}$ and is denoted by $v \oplus_r \mathcal{D}_{ex}^i$. Note that i may be negative in both cases. For relative neighbors we always have $v = v \oplus_r \mathcal{D}_{ex}^0$. In the rest of this section we only refer to absolute neighbors if not mentioned otherwise. Relative neighbors will be used later in Section 6.2.

From the start point of $\mathcal{C}_{ex}$, the robot explores a nonsimple path $\mathcal{P}_{ex}$ in $\hat{T}$ that shares the periodicity of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{R}_{cr}$. If the robot discovers irregularities in the exploration process, then it can use the irregularities to eliminate placements until the explored path is again periodic w.r.t. $\mathcal{R}_{cr}$.

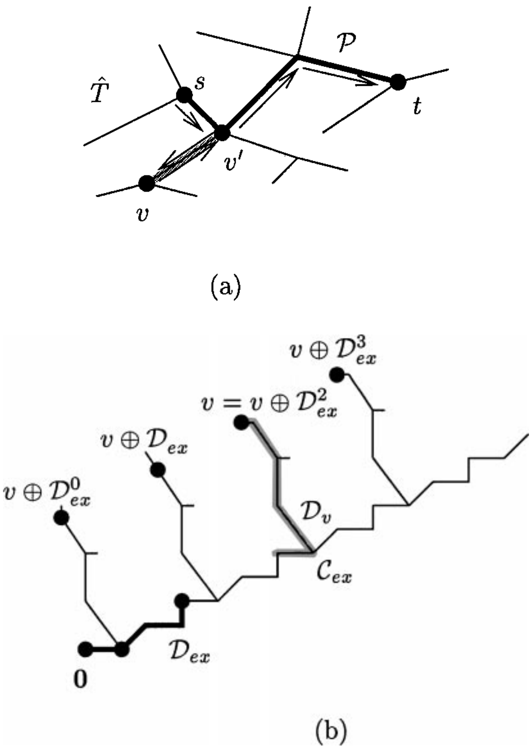
$\mathcal{P}$
$\hat{T}$
s
t
v'
v
(a)
$v \oplus \mathcal{D}_{ex}^3$
$v = v \oplus \mathcal{D}_{ex}^2$
$v \oplus \mathcal{D}_{ex}$
$v \oplus \mathcal{D}_{ex}^0$
$\mathcal{D}_v$
$\mathcal{C}_{ex}$
$\mathcal{D}_{ex}$
0
(b)

FIG. 9. Augmenting the path $\mathcal{P}$ by the point v: The robot follows the path $\mathcal{P}$ from s to the $\mathcal{P}$-base of v', travels to v and back, and then follows the rest of $\mathcal{P}$. (b) The absolute $\mathcal{D}_{ex}^k$-neighbors of v, for $0 \le k \le 3$, with $v = v \oplus \mathcal{D}_{ex}^2$. Note that $v \oplus \mathcal{D}_{ex}$ is not a vertex.

Initially, $\mathcal{P}_{ex} = \mathcal{C}_{ex}$. The robot augments $\mathcal{P}_{ex}$ by the vertices in $\hat{T}_{ex}^{S1}$. For each vertex v in $\hat{T}_{ex}^{S1}$, the robot visits the (absolute) $\mathcal{D}_{ex}^j$-neighbors u_j of v for $0 \le j \le \lfloor \lambda(\mathcal{C}_{ex})/\lambda(\mathcal{D}_{ex}) \rfloor$ and then updates P, $\mathcal{R}_{cr}$, and $\mathcal{D}_{ex}$ before examining the neighbors of the next vertex. It halts when either $\mathcal{R}_{cr}$ is empty or $\mathcal{P}_{ex}$ contains $\sqrt{n}$ vertices.

The strategy for extending the critical path to a periodic path containing $\sqrt{n}$ vertices can be described as follows.

```
Strategy Periodic-Path
Input:  $T$, $P$, and the explored tree $\hat{T}_{ex}^{S1}$;
Output: A periodic path $\mathcal{P}$ in $\hat{T}$ that shares the periodicity of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{R}_{cr}$ and contains at least
        $\sqrt{n}$ vertices or $\mathcal{R}_{cr} = \emptyset$;
$P_0 \leftarrow P$; $\mathcal{R}_0 \leftarrow \mathcal{R}_{cr}$; $\mathcal{P}_0 \leftarrow \mathcal{C}_{ex}$;
let $\mathcal{D}_0$ be the greatest common period of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{R}_0$;
let $\alpha_0$ be the periodicity of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{D}_0$; $k_0 \leftarrow \lfloor \alpha_0 \rfloor$; $\mathcal{C}_0 \leftarrow \mathcal{D}_0^{k_0}$;
let $v_0$ be the first unvisited vertex in $\hat{T}_{ex}^{S1}$;
$i \leftarrow 0$;
while $|\mathcal{P}_i| < \sqrt{n}$ and $\mathcal{R}_i \neq \emptyset$ do
    /* Invariant 1—see below */
    $\overline{\mathcal{P}_i} \leftarrow \mathcal{P}_i$;
    for $j \leftarrow 0$ to $k_i - 1$ do
        visit $u_j \leftarrow v_i \oplus \mathcal{D}_i^j$;
        augment $\overline{\mathcal{P}_i}$ by $u_j$;
    /* Update the variables: */
    $P_{i+1} \leftarrow \{p \in P_i \mid p + \hat{T}_{ex}^{S1} \subseteq T\}$;
    $\mathcal{R}_{i+1} \leftarrow \{(p, p') \in \mathcal{R}_i \mid p, p' \in P_{i+1}\}$;
    let $\mathcal{D}_{i+1}$ be the greatest common period of $\mathcal{C}_i$ w.r.t. $\mathcal{R}_{i+1}$;
    let $\alpha_{i+1}$ be the periodicity of $\mathcal{C}_i$ w.r.t. $\mathcal{D}_{i+1}$; $k_{i+1} \leftarrow \lfloor \alpha_{i+1} \rfloor$; $\mathcal{C}_{i+1} = \mathcal{D}_{i+1}^{k_{i+1}}$;
```

```
        $\mathcal{P}_{i+1} \leftarrow$ the part of $\overline{\mathcal{P}}_i$ from the origin to the end point of $\mathcal{C}_{i+1}$;
        let $v_{i+1}$ be the next unvisited vertex in $\hat{T}^{S1}_{ex}$;
        $i \leftarrow i+1$;
    end while
    $\kappa \leftarrow i$;
    $P \leftarrow P_\kappa$; $\mathcal{R}_{cr} \leftarrow \mathcal{R}_\kappa$; $\mathcal{C}_{ex} \leftarrow \mathcal{C}_\kappa$; $\mathcal{D}_{ex} \leftarrow \mathcal{D}_\kappa$; $\mathcal{P}_{ex} \leftarrow \mathcal{P}_\kappa$;
    end Periodic-Path
```

5.1. The Correctness of Strategy Periodic-Path

We now prove the correctness of Strategy *Periodic-Path*. In order to do so we make use of the notation used in the algorithmic description of the strategy without explicitly defining it again. The while-loop has the following invariant.

INVARIANT 1. *k_i is the largest integer such that $\mathcal{D}_i^{k_i}$ is contained in $\mathcal{C}_{ex}$.*

Proof. The invariant clearly holds for $i = 0$. So assume that the invariant holds up to iteration $i-1 \geq 0$. Let k_i^* be the largest integer j such that $\mathcal{D}_i^j$ is contained in $\mathcal{C}_{ex}$. Clearly, $k_i^* \geq k_i$. There is some integer $n_i \geq 1$ with $\mathcal{D}_{i-1}^{n_i} = \mathcal{D}_i$ by Lemma 4.6. Hence, $\mathcal{D}_i^{k_i^*} = \mathcal{D}_{i-1}^{n_i k_i^*} \subseteq \mathcal{C}_{ex}$. Since k_{i-1} is the largest integer j such that $\mathcal{D}_{i-1}^j$ is contained in $\mathcal{C}_{ex}$ by the invariant, $\mathcal{D}_{i-1}^{n_i k_i^*} \subseteq \mathcal{D}_{i-1}^{k_{i-1}} = \mathcal{C}_{i-1}$. Hence, $\mathcal{D}_i^{k_i^*}$ is contained in $\mathcal{C}_{i-1}$ and $k_i \geq k_i^*$ which proves the claim. ■

We now argue that Strategy *Periodic-Path* halts. Since $\mathcal{R}_i \neq \emptyset$, for $0 \leq i \leq \kappa - 1$, the following result is a direct consequence of Observation 4.3. For $0 \leq i \leq \kappa - 1$,

$$\lambda(\mathcal{D}_i) \leq \lambda(\mathcal{D}_{cr}). \tag{1}$$

Since $\lambda(\mathcal{C}_{ex}) = 4\lambda(\mathcal{D}_{cr})$, Eq. (1) and Invariant 1 imply that, for all $0 \leq i \leq \kappa - 1$,

$$3\lambda(\mathcal{D}_{cr}) \leq \lambda\left(\mathcal{D}_i^{k_i}\right) = \lambda(\mathcal{C}_i). \tag{2}$$

By the definition of $\mathcal{C}_{ex}$ as $\mathcal{D}_{cr}^4$ only the second and third quarter of $\mathcal{C}_{ex}$ may intersect $\hat{T}^{S1}_{ex}$. Since $\mathcal{C}_i$ contains the first three quarters of $\mathcal{C}_{ex}$ by Inequality (2), the $\mathcal{C}_{ex}$-base of vertex v_i belongs to $\mathcal{C}_i$. Hence, there is a $0 \leq j \leq k_i - 1$ such that $v_i \oplus \mathcal{D}_i^j$ equals v_i and the number of visited vertices increases by at least one in each iteration of the while-loop. Thus, we have shown the following lemma.

LEMMA 5.1. *The number of iterations κ of the outer while-loop is bounded by $\sqrt{n}$.*

At the end of Strategy *Periodic-Path*, $\mathcal{C}_{ex}$ is set to $\mathcal{C}_\kappa$. Inequality (2) now implies the following result.

LEMMA 5.2. *If $\mathcal{R}_\kappa \neq \emptyset$, then $\lambda(\mathcal{C}_{ex}) \geq 3\lambda(\mathcal{D}_{cr})$ after Strategy Periodic-Path.*

We will show that after each iteration of the while-loop the path $\mathcal{P}_i$ is periodic w.r.t. $\mathcal{R}_i$. Moreover, if $\mathcal{Q}_i$ is the greatest common period of $\mathcal{P}_i$ w.r.t. $\mathcal{R}_i$, then $\lambda_{sp(\mathcal{Q}_i)} = \lambda(\mathcal{D}_i)$. We first show the existence of a greatest common period of $\mathcal{P}_i$ w.r.t. $\mathcal{R}_i$. Let (p, p') be a critical pair in $\mathcal{R}_i$ with $v = p' - p \in \mathcal{C}_i$ and $\mathcal{P}_v$ the part of $\mathcal{P}_i$ from its starting point to v.

LEMMA 5.3. *There is an $\alpha > 1$ with $\mathcal{P}_v^\alpha = \mathcal{P}_i$.*

Proof. Let $\mathcal{C}_v$ be the shortest path from the origin to v. By Observation 4.2 there is an $\alpha > 1$ such that $\mathcal{C}_v^\alpha = \mathcal{C}_i$. Since $\mathcal{P}_v$ contains $\mathcal{C}_v$, $\mathcal{P}_v^\alpha$ reaches the end point of $\mathcal{C}_i$. We claim that $\mathcal{P}_v^\alpha = \mathcal{P}_i$. The proof is by contradiction. Assume that u is the first point on $\mathcal{P}_i$ where $\mathcal{P}_v^\alpha$ and $\mathcal{P}_i$ differ. Since $\mathcal{P}_v$ is an initial part of $\mathcal{P}_i$, u occurs after v on $\mathcal{P}_i$.

Let $u' = u - v$. Note that u' is reached before u by both $\mathcal{P}_v^\alpha$ and $\mathcal{P}_i$. There are two possible cases why $\mathcal{P}_v^\alpha$ and $\mathcal{P}_i$ may differ at u. Either u' is not locally isomorphic to u in $\mathcal{P}_i$, that is, not all edges of $\mathcal{P}_i$ incident to u' have the same orientations as the edges of $\mathcal{P}_i$ incident to u (note that in $\mathcal{P}_v^\alpha$ u' is clearly locally isomorphic to u), or there is an edge e' incident to u' which is explored at u' for some distance but its translate at u is not explored (or vice versa).

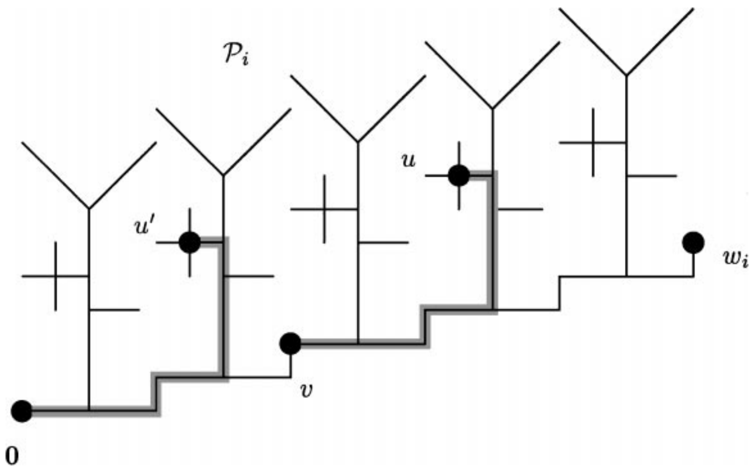

FIG. 10. The path $sp_{\hat{T}}(\mathbf{0}, u')$ is isomorphic to the path $sp_{\hat{T}}(v, u)$ in $\mathcal{P}_i$.

We first show that u' is locally isomorphic to u. Since p and p' are placements, $p + sp_{\hat{T}}(\mathbf{0}, u')$ is isomorphic to $p' + sp_{\hat{T}}(\mathbf{0}, u')$ in T. Furthermore, since v corresponds to the location of p' if p is mapped to the origin, $p' + sp_{\hat{T}}(\mathbf{0}, u')$ is isomorphic to $v + sp_{\hat{T}}(\mathbf{0}, u') = sp_{\hat{T}}(v, u)$ in $\mathcal{P}_i$. Therefore, $sp_{\hat{T}}(v, u)$ is isomorphic to the explored part of $sp_{\hat{T}}(\mathbf{0}, u')$ in $\mathcal{P}_i$ (see Fig. 10). The path $sp_{\hat{T}}(\mathbf{0}, u')$ is completely explored in $\mathcal{P}_i$ since otherwise the first point on $\mathcal{P}_i$ where $\mathcal{P}_v^\alpha$ and $\mathcal{P}_i$ differ is before u. Since $sp_{\hat{T}}(v, u)$ is clearly isomorphic to $sp_{\hat{T}}(\mathbf{0}, u')$ in $\mathcal{P}_v^\alpha$, $\mathcal{P}_i$ and $\mathcal{P}_v^\alpha$ are locally isomorphic at u.

Now suppose there is a (partially) explored edge e' that is incident to u', but the edge e incident to u that has the same orientation as e' is not explored. Let t' be the end point of the explored part of e' and $1 \leq j \leq i$ be the first iteration such that t' belongs to the path from the origin to a $\mathcal{D}$-neighbor of v_j.

We show that $t' + v$ belongs to the path from the origin to a neighbor $v_j \oplus \mathcal{D}_j^l$, for some $1 \leq l \leq k_j - 1$. Note that $t' + v$ is a $\mathcal{C}_v$-neighbor of t'. Since $1 \leq j \leq i$, Observation 4.2 and Lemma 2.3 imply that there is an integer l with $\mathcal{D}_j^l = \mathcal{C}_v$. Thus, since t' is visited in iteration j, all the $\mathcal{C}_v$-neighbors of t' in $\mathcal{P}_i$ are also visited in iteration j. In particular, $t' \oplus \mathcal{D}_j^l$ is visited in iteration j since $\mathcal{C}_i \subseteq \mathcal{C}_j$ and the $\mathcal{C}_{ex}$-base of $t' \oplus \mathcal{D}_j^l$ equals the $\mathcal{C}_{ex}$-base of u which belongs to $\mathcal{C}_i$.

Hence, there is an edge e incident to u that is partially explored and isomorphic to e'—a contradiction. If there is a partially explored edge e incident to u that is not explored at u' a similar argument applies. Since we reach a contradiction for either case, the claim follows. ■

Lemmas 2.3 and 5.3 now lead to the following corollary.

COROLLARY 5.1. *A greatest common period $\mathcal{Q}_i$ of $\mathcal{P}_i$ w.r.t. $\mathcal{R}_i$ exists and is well defined, for all $0 \leq i \leq \kappa$.*

Proof. Let $(p, p') \in \mathcal{R}_i$, $v = p' - p$, $\mathcal{C}_v$ be the shortest path from the origin to v, and $\mathcal{P}_v$ be the part of $\mathcal{P}_i$ from the origin to v. The claim follows from Lemma 2.3 if we can show that $\mathcal{P}_v^2$ is contained in $\mathcal{P}_i$.

By the definition of a critical pair, $\mathcal{C}_v$ is contained in $\mathcal{D}_{cr}$. Inequality (2) now implies that $\mathcal{C}_v^2$ is a subpath of $\mathcal{D}_i^{k_i}$.

Let v' be the end point of $\mathcal{C}_v^2$. v' also equals the end point of $\mathcal{P}_v^2$. Since $\mathcal{P}_v^\alpha = \mathcal{P}_i$, for some $\alpha > 1$, and $\mathcal{P}_i$ contains $\mathcal{D}_i^{k_i}$, $\mathcal{P}_v^2$ is also contained in $\mathcal{P}_i$ and the claim now follows by Lemma 2.3. ■

In the following let $\mathcal{Q}_i$ be the greatest common period of $\mathcal{P}_i$ w.r.t. $\mathcal{R}_i$. Next we show that the shortest path $\mathcal{E}_i$ from the start point to the end point of $\mathcal{Q}_i$ equals $\mathcal{D}_i$.

LEMMA 5.4. *If $\mathcal{E}_i$ is the shortest path from the start point to the end point of $\mathcal{Q}_i$, then $\mathcal{D}_i = \mathcal{E}_i$, for all $0 \leq i \leq \kappa$.*

Proof. If (p, p') is a critical pair and $v = p' - p$, then $\mathcal{E}_i$ is an integral period of the shortest path from the origin to v since $\mathcal{Q}_i$ is an integral period of $\mathcal{P}_v$ by the definition of $\mathcal{Q}_i$. By Lemma 2.3 this implies that there is a $j \geq 1$ with $\mathcal{E}_i^j = \mathcal{D}_i$. Hence, if (p, p') is a critical pair and $v = p' - p$, then $\mathcal{Q}_i^j$ is also an integral period of $\mathcal{P}_v$. Since $\mathcal{Q}_i$ is a greatest common period w.r.t. $\{\mathcal{P}_v \mid v = p' - p$ where $(p, p') \in \mathcal{R}_{cr}\}$, $j = 1$ and $\mathcal{E}_i = \mathcal{D}_i$ as claimed. ■

Hence, after the while-loop is exited either the path $\mathcal{P}_\kappa$ is periodic w.r.t. $\mathcal{R}_{cr}$ by Corollary 5.1 and contains at least $\sqrt{n}$ vertices or $\mathcal{R}_{cr} = \emptyset$. Together with the observation that the loop halts, this proves the correctness of Strategy *Periodic-Path*.

5.2. Analysis of Strategy *Periodic-Path*

In the following we investigate how far the robot travels during the execution of Strategy *Periodic-Path*. We first investigate how far the robot travels to visit $\mathcal{D}_i$-neighbors of v_i that are vertices. Since the while-loop is exited once $\mathcal{P}_i$ contains $\sqrt{n}$ vertices, the total number of visited neighbors that are vertices for iterations $0 \le i < \kappa - 1$ is $\sqrt{n}$. In the last iteration $\kappa - 1$ the robot visits at most as many $\mathcal{D}_{\kappa-1}$-neigbors of $v_{\kappa-1}$ as there are vertices in $\mathcal{C}_{ex}$. By Lemma 4.5 the number of vertices in $\mathcal{C}_{ex}$ is $O(\sqrt{n})$. As the distance to visit a $\mathcal{D}_i$-neighbor of v_i is bounded by the diameter of $\mathcal{C}_{ex} \cup \hat{T}_{ex}^{S1}$ which is $O(\varrho)$ we have shown the following lemma.

LEMMA 5.5. *The distance traveled by the robot to visit all $\mathcal{D}_i$-neighbors of v_i that are vertices summed over all iterations* $0 \le i \le \kappa - 1$ *is* $O(\sqrt{n}\varrho)$.

Not necessarily all of the $\mathcal{D}_i$-neighbors of v_i are vertices; in fact, none of them may be a vertex—except for v_i itself. Yet the robot may have to travel a distance of $\Theta(\varrho)$ for each of the $\mathcal{D}_i$-neighbors of v_i. But we can show that the total number of nonvertex $\mathcal{D}_i$-neighbors of v_i visited by the robot is also bounded by $O(\sqrt{n})$.

LEMMA 5.6. *If f_i is the fraction of neighbors $v_i \oplus \mathcal{D}_i^j$ that are vertices of $\hat{T}$, then $\lambda_{sp(\mathcal{Q}_{i+1})} \ge \lceil 1/f_i \rceil\, \lambda_{sp(\mathcal{Q}_i)}$, for all* $0 \le i \le \kappa - 1$.

Proof. By Lemmas 5.4 and 4.6 there exists an integer m_i such that $\mathcal{Q}_i^{m_i} = \mathcal{Q}_{i+1}$ and $\mathcal{D}_i^{m_i} = \mathcal{D}_{i+1}$. Since, for each $0 \le j \le k_i - 1$, there is a $\mathcal{D}_i$-neighbor of v_i that is visited by the robot and $\mathcal{Q}_{i+1}$ contains m_i $\mathcal{D}_i$-neighbors of v_i by Lemma 5.4, the number of $\mathcal{D}_i^j$-neighbors of v_i that are vertices in $\mathcal{Q}_{i+1}$ is exactly $f_i m_i$. There is at least one vertex in $\mathcal{Q}_{i+1}$ that is isomorphic to v_i, since $\mathcal{Q}_{i+1}$ is a period of $\mathcal{P}_{i+1}$ and $\mathcal{P}_{i+1}$ contains v_i. Therefore, $f_i m_i \ge 1$ or $m_i \ge 1/f_i$. As m_i is an integer we have $m_i \ge \lceil 1/f_i \rceil$. ■

LEMMA 5.7. *Summed over all iterations i, the robot visits a total of at most $O(\sqrt{n})$ $\mathcal{D}_i$-neighbors of v_i that are not vertices.*

Proof. The number k_0 of $\mathcal{D}_0$-neighbors visited in the first iteration is at most $12\sqrt{n}$ since $\hat{T}_{ex}^{S1}$ contains $\sqrt{n}$ vertices and the length of $\mathcal{C}_{ex}$ is at most 12 times the diameter of $\hat{T}_{ex}^{S1}$. If k_i neighbors are visited in iteration i of which $f_i k_i$ are vertices, then $\lambda_{sp(\mathcal{Q}_{i+1})} \ge \lceil 1/f_i \rceil\, \lambda_{sp(\mathcal{Q}_i)}$ by Lemma 5.6. Hence, the number of neighbors visited in iteration $i+1$ is at most $k_i / \lceil 1/f_i \rceil$ and the number of nonvertices visited is at most $(1 - f_{i+1})k_i / \lceil 1/f_i \rceil$. Note that this is 0 if $f_{i+1} = 1$. So let $i_1, \ldots, i_k$ be the iterations for which $f_{i_j} < 1$. Hence, the total number of visited nonvertices is no more than

$$k_0 \sum_{j=1}^{k} \frac{(1 - f_{i_j})}{\prod_{l=1}^{j-1} \lceil 1/f_{i_l} \rceil}.$$

Since $\lceil 1/f_{i_j} \rceil \ge 2$, the above term is bounded by

$$k_0 \sum_{j=1}^{k} \frac{1}{2^{j-1}} < 2k_0 \le 24\sqrt{n}$$

which proves the claim. ■

Since the distance of a $\mathcal{D}_i$-neighbor of v_i is $O(\varrho)$, we can summarize the situation at the end of Strategy *Periodic-Path* as follows.

LEMMA 5.8. *The robot travels a distance of $O(\sqrt{n}\varrho)$ during the execution of Strategy* Periodic-Path, *and after executing it either there are no critical pairs left or it has found a path in $\hat{T}$ that is periodic w.r.t. $\mathcal{R}_{cr}$ and contains at least $\sqrt{n}$ vertices of $\hat{T}$ and at most $24\sqrt{n}$ leaves that are not vertices of $\hat{T}$.*

6. STEP 4: EXTENDING THE CRITICAL PATH

If T were infinite then all placements in P could be the wake-up position after Step 3 and the periodic path $\mathcal{P}_{ex}$ could continue to infinity. But T is finite. Hence, if we concatenate together enough copies of $\mathcal{P}_{ex}$ then there will be many differences—or *mismatches*—between T and the copies of $\mathcal{P}_{ex}$. In the following we show how to make use of these differences.

6.1. Identifying an Initial Mismatch

Let $\mathcal{P}_{ex}^*$ be the periodic path that is formed by concatenating together $\lceil 2\sqrt{n}\rceil$ copies of $\mathcal{P}_{ex}$. Note that the diameter of $\mathcal{P}_{ex}^*$ is $O(\sqrt{n}\lambda(\mathcal{D}_{cr})) = O(\sqrt{n}\varrho)$. The tree $\hat{T}_{ex}^*$ induced by $\mathcal{P}_{ex}^*$ contains at least $2n$ vertices and, hence, there are at least n vertices where $\hat{T}_{ex}^*$ differs from $\hat{T}$. In the following we show that the robot can find an initial mismatch q^* by exploring at most $\log|P| \leq \log n$ vertices of $\mathcal{P}_{ex}^*$. Once q^* is found, the robot uses q^* as a seed mismatch to find mismatches that are closer and closer to $\mathbf{0}$ until $\mathcal{R}_{cr} = \emptyset$.

LEMMA 6.1. *There is a vertex v in $\hat{T}_{ex}^*$ such that $p + sp_{\hat{T}_{ex}^*}(\mathbf{0}, v) \not\subseteq T$, for at least one half of the placements p in P.*

Proof. Let W be the set of all pairs (p, v) with $p \in P$ and $v \in \hat{T}_{ex}^*$ such that $p + v \notin T$. For each $p \in P$, we denote the set of pairs in W with first component p by W_p. Since $|W_p| \geq |\hat{T}_{ex}^*| - |T|$, the sum over the cardinalities of all W_p is given by

$$|W| = \sum_{p\in P} |W_p| \geq \sum_{i=1}^{|P|} (|\hat{T}_{ex}^*| - |T|) = |P|\,(|\hat{T}_{ex}^*| - |T|).$$

If W_v is the set of pairs with second component v, then $\sum_{v\in\hat{T}_{ex}^*} |W_v| = |W| \geq |P|\,(|\hat{T}_{ex}^*| - |T|)$. Hence, the average size of the $|\hat{T}_{ex}^*|$ sets W_v is at least

$$\frac{|P|\,(|\hat{T}_{ex}^*| - |T|)}{|\hat{T}_{ex}^*|} = |P|\left(1 - \frac{|T|}{|\hat{T}_{ex}^*|}\right) \geq \frac{|P|}{2}.$$

Therefore, there is at least one vertex $v \in \hat{T}_{ex}^*$ with $|W_v| \geq |P|/2$. ■

If we apply Lemma 6.1 repeatedly and remove each time the set of placements p from P for which a vertex v in $\hat{T}_{ex}^*$ is found such that $p + sp_{\hat{T}_{ex}^*}(\mathbf{0}, v) \not\subseteq T$, then we obtain the following corollary.

COROLLARY 6.1. *There exists a set V_0 of $\lceil\log|P|\rceil$ vertices in $\hat{T}_{ex}^*$ such that, for each $p \in P$, there is a vertex $v \in V_0$ such that $p + sp_{\hat{T}_{ex}^*}(\mathbf{0}, v) \not\subseteq T$.*

Note that the vertices of Corollary 6.1 can be computed given T and $\hat{T}_{ex}^*$ without any additional exploration by the robot.

In the following let $\mathcal{C}_{ex}^*$ be the path formed by concatenating $2\lceil\sqrt{n}\rceil$ copies of $\mathcal{C}_{ex}$. We sort the vertices in V_0 according to their distance of their $\mathcal{C}_{ex}^*$-bases to the origin. The robot travels along $\mathcal{C}_{ex}^*$ visiting the vertices in V_0 in sequence until the first vertex $v^* \in \hat{T}_{ex}$ is identified for which $sp_{\hat{T}}(\mathbf{0}, v^*) \not\subseteq \hat{T}$.

Note that the distance of each vertex $v \in V_0$ to $\mathcal{C}_{ex}^*$ is at most 4ϱ; hence, the robot travels a distance of at most $\lambda(\mathcal{C}_{ex}^*) + 2\log n\, 4\varrho = O(\sqrt{n}\varrho)$ in order to identify v^*. Let q^* be the last common point of $sp_{\hat{T}_{ex}^*}(\mathbf{0}, v^*)$ and $\hat{T}$. The point q^* is the initial mismatch the robot is looking for.

6.2. Strategy Mismatch-Propagation

Once we have identified the mismatch q^*, it is our aim to find the closest mismatch to $\mathcal{P}_{ex}$. Before we describe how to do so we need one more definition. Recall that $\mathcal{D}_{ex}$ is the critical period of $\mathcal{C}_{ex}$. We say a point q' in $\hat{T}$ is $\mathcal{D}_{ex}$*-isomorphic* to a point q in $\hat{T}$ if there is an integer j such that $q' = q \oplus_r \mathcal{D}_{ex}^j$ and q' is the end point of the path $q + \mathcal{D}_{ex}^j$.[4]

[4] Here we make use of the definition of relative neighbor.

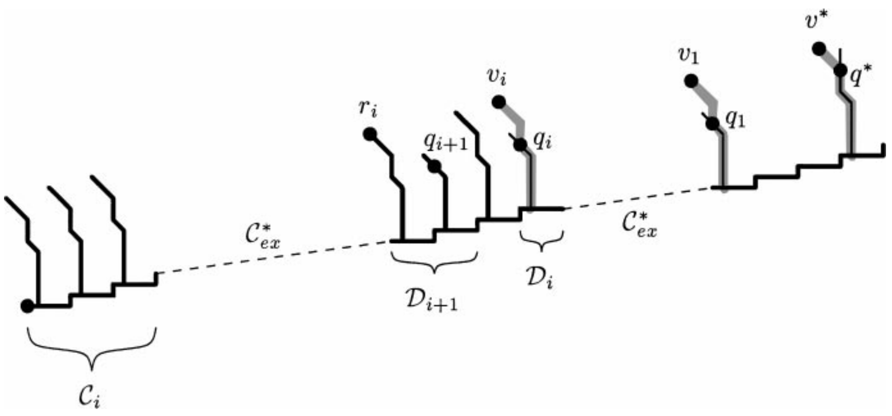

FIG. 11. Since $r_i = v_i \oplus_r \mathcal{C}_i^{-1}$ is $\mathcal{D}_{ex}$-isomorphic to v^*, all $\mathcal{D}_i$-neighbors of r_i between r_i and q_i are explored. This yields the new point q_{i+1} and at least doubles the length of the greatest common period $\mathcal{D}^i_{ex}$ of $\mathcal{C}_i$ w.r.t. $\mathcal{R}_i$.

The robot now returns to the origin and on its way it visits points that are not $\mathcal{D}_{ex}$-isomorphic to v^*, with q^* being the first such point. It does so by looking at the $\mathcal{C}_{ex}$-neighbors of v^* that are closer to the origin. More precisely, the robot computes a sequence of points $q_i \in \hat{T}$ and $v_i \in \hat{T}^*_{ex}$, where v_i is $\mathcal{D}_{ex}$-isomorphic to v^* and q_i is not. To compute v_{i+1} and q_{i+1}, given v_i and q_i, the robot first visits the point $r_i = v_i \oplus_r \mathcal{C}_{ex}^{-1}$.

There are two cases. If r_i is not $\mathcal{D}_{ex}$-isomorphic to v^* (like q^*), then the robot has found a mismatch that is closer to $\mathcal{P}_{ex}$, sets $q_{i+1} = r_i$, v_{i+1} to the end point of the path $v_i + \mathcal{C}_{ex}^{-1}$, and continues.

Otherwise, r_i is $\mathcal{D}_{ex}$-isomorphic to v^*. The robot now explores all the $\mathcal{D}_{ex}$-neighbors of r_i between r_i and q_i. This is exactly analogous to one iteration of Strategy *Periodic-Path*. After the exploration of the $\mathcal{D}_{ex}$-neighbors the robot recomputes the critical pairs $\mathcal{R}_{cr}$. Since r_i is $\mathcal{D}_{ex}$-isomorphic to v^* but q_i is not, the greatest common period $\mathcal{D}_{ex}$ of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{R}_{cr}$ changes; this implies that some pairs in $\mathcal{R}_{cr}$ are eliminated. Among the explored $\mathcal{D}_{ex}$-neighbors of v^* between r_i and q_i we choose q_{i+1} to be the closest one to the origin that is not $\mathcal{D}_{ex}$-isomorphic to v^*. We are done if $\mathcal{R}_{cr} = \emptyset$.

The strategy of the robot can now be described as follows. Figure 11 illustrates the algorithm.

```
Strategy Mismatch-Propagation
Input:  The tree $T$, the set of placements $P$, the periodic paths $\mathcal{P}_{ex}$, $\mathcal{C}_{ex}$, and the points $q^*$ and $v^*$;
Output: A set of placements $P$ such that $\mathcal{R}_{cr} = \emptyset$;
 1.  $P_0 \leftarrow P$; $\mathcal{R}_0 \leftarrow \mathcal{R}_{cr}$; $q_0 \leftarrow q^*$; $v_0 \leftarrow v^*$;
 2.  let $\mathcal{D}_0$ be the greatest common period of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{R}_0$;
 3.  let $\alpha_0$ be the periodicity of $\mathcal{C}_{ex}$w.r.t. $\mathcal{D}_0$; $k_0 \leftarrow \lfloor \alpha_0 \rfloor$; $\mathcal{C}_0 \leftarrow \mathcal{D}_0^{k_0}$;
 4.  $i \leftarrow 0$;
 5.  while $\mathcal{R}_i \neq \emptyset$ do
         /* Invariant 2—see below */
 6.      visit $r_i \leftarrow v_i \oplus_r \mathcal{C}_i^{-1}$;
 7.      if $r_i$ is $\mathcal{D}_{ex}$-isomorphic to $v_i$ (and $v^*$)
 8.        then for $j \leftarrow 1$ to $k_i - 1$ do visit $r_i \oplus_r \mathcal{D}_i^j$;
 9.             let $j^*$ be the smallest index such that $r_i \oplus_r \mathcal{D}_i^{j^*}$ is not isomorphic to $r_i$;
10.             $q_{i+1} \leftarrow r_i \oplus_r \mathcal{D}_i^{j^*}$;
11.             $v_{i+1} \leftarrow$ the end point of the path $r_i + \mathcal{D}_i^{j^*}$;
12.        else $q_{i+1} \leftarrow r_i$;
13.             $v_{i+1} \leftarrow$ the end point of the path $v_i + \mathcal{C}_{ex}^{-1}$;
         /* Update the variables: */
14.      $P_{i+1} \leftarrow \{p \in P_i \mid p + \hat{T}_{ex} \subseteq T\}$;
15.      $\mathcal{R}_{i+1} \leftarrow \{(p, p') \in \mathcal{R}_i \mid p, p' \in P_{i+1}\}$;
16.      let $\mathcal{D}_{i+1}$ be the greatest common period of $\mathcal{C}_i$ w.r.t. $\mathcal{R}_{i+1}$;
```

```
17.       let $\alpha_{i+1}$ be the periodicity of $\mathcal{C}_i$ w.r.t. $\mathcal{D}_{i+1}$; $k_{i+1} \leftarrow \lfloor \alpha_{i+1} \rfloor$; $\mathcal{C}_{i+1} \leftarrow \mathcal{D}_{i+1}^{k_{i+1}}$;
18.       $i \leftarrow i + 1$;
      end while
19.   $P \leftarrow P_i$;
      end Mismatch-Propagation
```

6.3. The Correctness of Strategy Mismatch-Propagation

We only have to show that the algorithm halts. In order to do so we make use of a potential function Φ and show that Φ is bounded from below and reduced by a constant amount in each iteration of the outer while-loop.

As in Section 5.1 we see that the following invariant holds for the while-loop of Strategy *Mismatch-Propagation*.

INVARIANT 2. *k_i is the largest integer such that $\mathcal{D}_i^{k_i}$ is contained in $\mathcal{C}_{ex}$.*

The invariant again implies Inequality (2) which we state again for completeness. For all iterations i except the last one

$$3\lambda(\mathcal{D}_{cr}) \leq \lambda\big(\mathcal{D}_i^{k_i}\big) = \lambda(\mathcal{C}_i). \tag{2}$$

Let d_i be the distance of the $\mathcal{C}_{ex}^*$-base q_i^* of q_i to the origin; that is, $d_i = d_{\hat{T}}(\mathbf{0}, q_i^*)$. The potential Φ_i in iteration i is given by

$$\Phi_i = d_i - \frac{\lambda(\mathcal{D}_i)}{\lambda(\mathcal{D}_0)}\lambda_{cr},$$

where $\lambda_{cr} = \lambda(\mathcal{D}_{cr})$. We first show that Φ_i is reduced in each iteration by at least λ_{cr}.

LEMMA 6.2. *For all iterations i,*

$$\Phi_i - \Phi_{i+1} \geq \lambda_{cr}.$$

Proof. Consider iteration i of the outer loop. If r_i is not $\mathcal{D}_{ex}$-isomorphic to v_i, then $q_{i+1} = r_i$ and q_{i+1}^* is a distance of $\lambda(\mathcal{C}_i)$ closer to the origin than q_i^*. Hence, $\Phi_i - \Phi_{i+1} \geq d_i - d_{i+1} = \lambda(\mathcal{C}_i) \geq 3\lambda(\mathcal{D}_{cr})$ and Φ_i is reduced by at least λ_{cr} in this case.

Now assume that r_i is $\mathcal{D}_{ex}$-isomorphic to v_i. Let m_i be the integer such that $\mathcal{D}_i^{m_i+1}$ contains the $\mathcal{C}_{ex}^*$-base of r_i, u_{m_i} the end point of $\mathcal{D}_i^{m_i}$, and U_{i+1} the set of p-vertices v that are given by $v = p' - p$ with $(p, p') \in \mathcal{R}_{i+1}$. Furthermore, let $\mathcal{P}_{i+1}$ be the path starting at u_{m_i} to the end point of $\mathcal{D}_i^{m_i+k_i}$ augmented by the $\mathcal{D}_i$-neighbors of r_i.

If the vertex v is in $u_{m_i} + U_{i+1}$, then the part $\mathcal{P}_v$ of $\mathcal{P}_{i+1}$ from u_{m_i} to v is a period of $\mathcal{P}_{i+1}$ by Lemma 5.3. Since $\lambda_{sp(\mathcal{P}_v)} \leq \lambda_{cr}$ and $\lambda(\mathcal{C}_i) \geq 3\lambda_{cr}$, we obtain that the period of $\mathcal{P}_{i+1}$ w.r.t. $\mathcal{P}_v$ is at least two as in the proof of Corollary 5.1; hence, a greatest common period $\mathcal{Q}_{i+1}$ of $\mathcal{P}_{i+1}$ w.r.t. $\mathcal{R}_{i+1}$ exists and is well defined. As in Section 5.1 it can be seen that $\mathcal{D}_{i+1}$ is the shortest path from the start point of $\mathcal{Q}_{i+1}$ to the end point of $\mathcal{Q}_{i+1}$.

As q_{i+1} is the closest $\mathcal{D}_i$-neighbor of r_i that is not $\mathcal{D}_{ex}$-isomorphic to v_i and $\mathcal{Q}_{i+1}$ is a period of the path $\mathcal{P}_{i+1}$, $\mathcal{Q}_{i+1}$ contains both r_i and q_{i+1}. Moreover, since the $\mathcal{C}_{ex}^*$-bases of r_i and q_{i+1} are separated at least by $\mathcal{D}_i$, $\lambda_{sp(\mathcal{Q}_{i+1})} > \lambda(\mathcal{D}_i)$. Hence, by Lemmas 4.6 and 5.4 there is an integer $k > 1$ such that $\mathcal{D}_{i+1} = \mathcal{D}_i^k$. Therefore, $\lambda(\mathcal{D}_i)/\lambda(\mathcal{D}_0) - \lambda(\mathcal{D}_{i+1})/\lambda(\mathcal{D}_0) = (k-1)\lambda(\mathcal{D}_{i+1})/\lambda(\mathcal{D}_0)$ and Φ_i is reduced by at least $(k-1)\lambda_{cr} \geq \lambda_{cr}$. ■

Note that Φ_i is at most $\lambda(\mathcal{C}_{ex}^*) = O(\sqrt{n}\lambda_{cr})$ in the beginning. We show that there is a lower bound for Φ_i, that is, that d_i is bounded from below and $\lambda(\mathcal{D}_i)/\lambda(\mathcal{D}_0)$ is bounded from above. The latter is easy to see since $\mathcal{D}_i$ is at most as long as $\mathcal{C}_{ex}$ and the periodicity of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{D}_0$ is bounded by the number of vertices in $\mathcal{C}_{ex}$ which is $O(\sqrt{n})$. In the next lemma we show that d_i is nonnegative which implies that Φ_i is bounded from below by $\Omega(-\sqrt{n}\lambda_{cr})$.

LEMMA 6.3. *For all iterations i, $d_i \geq 0$.*

Proof. Note that by definition $\lambda(\mathcal{C}_i) \leq \lambda(\mathcal{C}_{ex})$. Hence, the robot cannot skip over the periodic path $\mathcal{P}_{ex}$ in one iteration. So assume that there is an iteration l such that r_l belongs to $\mathcal{P}_{ex}$; otherwise, d_i clearly remains positive. We show that in this case $\mathcal{R}_{l+1} = \emptyset$ and l is the last iteration of the while-loop by Observation 4.1.

Since r_l belongs to $\mathcal{P}_{ex}$, it is $\mathcal{D}_{ex}$-isomorphic to v^* (since $\mathcal{P}_{ex}$ is periodic w.r.t. $\mathcal{D}_{ex}$ and v^* belongs to $\mathcal{P}^*_{ex}$). Let $\mathcal{P}_l$ be the path followed by the robot in iteration l. Since $\mathcal{D}_{ex}$ is an integral period of $\mathcal{D}_l$ by Lemma 4.6, all $\mathcal{D}_l$-neighbors of r_l that are visited in iteration l and belong to $\mathcal{P}_{ex}$ are $\mathcal{D}_{ex}$-isomorphic to v^*. Hence, q_{l+1} occurs after the end of $\mathcal{P}_{ex}$.

Now assume that $\mathcal{R}_{l+1} \neq \emptyset$ and let $(p, p') \in \mathcal{R}_{l+1}$. By the definition of $\mathcal{D}_l$ there is a $k \geq 1$ such that $\mathcal{D}_l^k = sp_{\hat{T}}(\mathbf{0}, v)$ where $v = p' - p$.

Since $q_{l+1} = r_l \oplus_r \mathcal{D}_l^{j^*}$, all vertices $r_l \oplus_r \mathcal{D}_l^j$ with $0 \leq j < j^*$ are $\mathcal{D}_{ex}$-isomorphic to v^*. Since q_{l+1} does not belong to $\mathcal{P}_{ex}$ but v belongs to $\mathcal{C}_{ex} \subseteq \mathcal{P}_{ex}$, $j^* > k$.

Let $r^* = r_l \oplus_r \mathcal{D}_l^{j^*-k}$. r^* is $\mathcal{D}_{ex}$-isomorphic to v^*. Consider the shortest path $sp_{\hat{T}}(\mathbf{0}, r^*)$ from the origin to r^*. Since p' is a placement, the path $p' + sp_{\hat{T}}(\mathbf{0}, r^*)$ belongs to T. Moreover, since p is a placement and v belongs to $\mathcal{C}_{ex}$, the path

$$(p + sp_{\hat{T}}(\mathbf{0}, v)) * (p' + sp_{\hat{T}}(\mathbf{0}, r^*)) = p + (sp_{\hat{T}}(\mathbf{0}, v) * sp_{\hat{T}}(\mathbf{0}, r^*)) = p + sp_{\hat{T}}(\mathbf{0}, v + r^*)$$

is contained in T and its end point is $\mathcal{D}_{ex}$-isomorphic to v^*. But, then $v + r^* \neq q_{l+1}$ is the $\mathcal{D}_l^{j^*}$-neighbor of r_l, a contradiction. ■

Lemmas 6.3 and 6.2 together with the upper bound on Φ_0 of $O(\sqrt{n}\lambda_{cr})$ imply that Strategy *Mismatch-Propagation* halts. More precisely, we have the following result.

COROLLARY 6.2. *The number of iterations of the outer loop is bounded by* $O(\sqrt{n})$.

6.4. Analysis of Strategy Mismatch-Propagation

Now we consider the distance traveled by the robot. We divide the distance the robot travels into four parts.

1. The distance δ_1 that the robot travels along $\mathcal{C}^*_{ex}$ from q^* toward the origin,
2. the distance δ_2 that the robot travels to visit the $\mathcal{C}_i$-neighbors in Step 6,
3. the distance δ_3 that the robot travels on $\mathcal{C}^*_{ex}$ in Step 8, and, finally,
4. the distance δ_4 that the robot travels to visit the $\mathcal{D}_i$-neighbors in Step 8.

The distance δ_1 is bounded by $O(\sqrt{n}\varrho) + \delta_3$ since the robot always travels toward the origin except in Step 8 in which it travels toward q^*. Since the length of $\mathcal{C}^*_{ex}$ is bounded by $O(\sqrt{n}\varrho)$ and each time the robot travels on $\mathcal{C}^*_{ex}$ toward q^* in Step 8, it travels at most the same amount toward the origin again, the bound on δ_1 follows.

Since the number of iterations of the outer loop of Strategy *Mismatch-Propagation* is bounded by $O(\sqrt{n})$ by Corollary 6.2 and the robot visits exactly one $\mathcal{C}_i$-neighbor at a distance of at most 4ϱ to $\mathcal{C}^*_{ex}$ in each iteration, the distance δ_2 is bounded by $O(\sqrt{n}\varrho)$. For the same reason the distance δ_3 is bounded by $O(\sqrt{n}\varrho)$ since in each iteration the robot travels at most a distance of $2\lambda(\mathcal{C}_i) = O(\varrho)$ toward q^*.

Finally, we bound the distance δ_4. Assume that in iteration i r_i is $\mathcal{D}_{ex}$-isomorphic to v_i in Step 7. As we observed in the proof of Lemma 6.2, this implies that $\mathcal{D}_{i+1} = \mathcal{D}_i^k$ for some integer $k > 1$; that is, the length of the critical period at least doubles, and, therefore, the periodicity of $\mathcal{C}_i$ w.r.t. $\mathcal{D}_{i+1}$ is at most half of the periodicity of $\mathcal{C}_{i-1}$ w.r.t. $\mathcal{D}_i$. Since $\mathcal{C}_{ex}$ contains at most $O(\sqrt{n})$ vertices, the periodicity of $\mathcal{C}_{ex}$ w.r.t. $\mathcal{D}_0$ $(=\mathcal{D}_{ex})$ is at most $O(\sqrt{n})$ and the robot executes the steps of the if-statement in Step 7 at most $O(\log\sqrt{n})$ times. As above, each $\mathcal{D}_i$-neighbor has a distance of at most 4ϱ to $\mathcal{C}^*_{ex}$ and, hence, we only need to estimate the number of $\mathcal{D}_i$-neighbors that are visited. Let the iterations in which the if-statement in Step 7 is true be $i_0, i_1, \ldots, i_m$ and k_{i_j} be the number of $\mathcal{D}_{i_j}$-neighbors that are visited in this iteration. We observed above that $\mathcal{D}_{i_{j+1}}$ is at least twice as long as $\mathcal{D}_{i_j}$. Hence, $k_{i_{j+1}} \leq k_{i_j}/2$, for

$0 \leq j \leq m-1$. Therefore, the total number of $\mathcal{D}_i$-neighbors that are visited is bounded by

$$\sum_{i=0}^{m} k_{i_j} \leq \sum_{j=0}^{m} k_{i_0} \left(\frac{1}{2}\right)^j \leq 2k_{i_0} \leq O(\sqrt{n})$$

since $k_{i_0} = O(\sqrt{n})$ as we observed before. Hence, $\delta_4 = O(\sqrt{n}\varrho)$.

This completes the analysis of Strategy *Mismatch-Propagation* and shows that the robot travels at most $O(\sqrt{n}\varrho)$ during its execution. Since $\mathcal{R}_{cr} = \emptyset$ at the end of Step 4, $p + \hat{T}_{ex}^{S1}$ contains no placements different from p by Lemma 4.4. We know therefore by Corollary 3.1 that $|P| \leq 2\sqrt{n}$, so we can continue with Step 5, the greedy strategy.

7. STEP 5: GREEDY ELIMINATION

We enter this step if the set of placements P (now again including the placements P_{triv} that we had excluded in Section 4.1) has size $|P| = O(\sqrt{n})$. In order to eliminate the remaining placements the robot visits the closest point q such that at least one placement is eliminated. Since $d_T(\mathbf{0}, q)$ is no more than the shortest distance to localize the robot, a repeated application of this procedure eventually eliminates all placements but one with a competitive ratio of at most $|P|$.

THEOREM 7.1 [7, 11]. *If $|P| = O(\sqrt{n})$ then the greedy strategy localizes the robot with a competitive ratio of $O(\sqrt{n})$.*

8. SEARCHING FOR A TARGET IN A TREE

As was pointed out in the Introduction, we are also interested in the searching variant of the problem where a robot has to find a target t marked on the map of the environment, but the robot is not given its wake-up position. We also apply Strategy LPS, except with a slightly altered Step 5. Note that after Step 1 if t is not reached, then the distance between the wake-up position s of the robot and t is at least ϱ. Hence, after Step 4 the robot has traveled a distance of $O(\sqrt{n}d_T(s,t))$ and there are $k \leq 2\sqrt{n}$ possible placements for the robot and thus k possible locations $t_1, \ldots, t_k$ of t in $\hat{T}$. In Step 5 the robot now repeatedly attempts to visit the closest possible target among $t_1, \ldots, t_k$ until the true location of the target is identified. Obviously, the robot travels at most a distance of $2\sqrt{n}d_T(s,t)$ in Step 5. This proves the following theorem.

THEOREM 8.1. *Let T be a geometric tree and t be a point in T. There is a strategy that achieves a competitive ratio of $O(\sqrt{n})$ for a robot to find t given the coordinates of t in T.*

9. IMPLEMENTATION

We have also implemented the algorithm, using `C++` and `LEDA` [12]. The algorithm is embedded in `OnVis`, a system for visualization of online algorithms developed at the MPI. As it turns out, a real robot should not run the algorithm exactly as described in the previous sections but instead it should use shortcuts at various points [8].

- Before starting each depth-first search in Step 1 the robot computes the explored tree. If there is no new information in distance d, which means no further placements can be eliminated by exploring the neighborhood within distance d, the robot leaves out the current distance d and continues with $2d$. If the computed tree has size $\sqrt{n}$ and at least $\sqrt{n}$ vertices lie in the ball of radius d, we can finish the spiral search.
- There may also be directions that provide no new information in the current distance or are already completely explored. For the robot it is needless to travel these paths.

- While computing a unique critical path the robot should not follow a path Q that is a subset of the explored tree $\hat{T}_{ex}$.
- In the Strategy *Periodic-Path* the robot can omit vertices that are already known.
- In the greedy elimination step the robot does not need to travel back from the closest point q_i to the origin, but moves directly to the next point q_{i+1} if it could not localize.
- In addition the testing of the explored tree $\hat{T}_{ex}$ on the placements could be done whenever we extend $\hat{T}_{ex}$.

10. CONCLUSIONS

We have presented a new localization strategy for an autonomous mobile robot. The environment of the robot is represented by a geometric tree of constant degree in arbitrary dimensions. We assume that the robot knows its current orientation and it has no use of vision other than to be able to detect the orientation of all edges incident to its current location. Our strategy, which solves an open problem posed by Kleinberg [11], achieves a competitive ratio of $O(\sqrt{n})$ if the tree contains n nodes of degree greater than or equal to three. Since there is a geometric tree that provides a lower bound of $\Omega(\sqrt{n})$ for the competitive ratio of any localization strategy, our strategy is optimal up to a constant factor. We also show that a slight modification of our strategy solves the problem of searching for a target in a geometric tree with the same competitive ratio.

Challenges that remain for future work are to transfer the strategies developed for geometric trees to the more realistic setting of polygons in the plane and to investigate the complexity of the localization problem in graph structures that allow cycles.

ACKNOWLEDGMENT

We thank the unknown referees for their helpful comments on an earlier version of this paper.

REFERENCES

1. Baeza-Yates, R., Culberson, J. C., and Rawlins, G. J. E. (1993), Searching in the plane, *Inform. and Comput.* **106**, 234–252.
2. Bar-Eli, E., Berman, P., Fiat, A., and Yan, P. (1994), Online navigation in a room, *J. Algorithms* **17**, 319–341.
3. Berman, P. (1998), On-line searching and navigation, *in* "Online Algorithms: The State of the Art" (A. Fiat and G. J. Woeginger, Eds.), pp. 232–241, Springer-Verlag, Berlin.
4. Betke, M., Rivest, R. L., and Singh, M. (1995), Piecemeal learning of an unknown environment, *Machine Learning* **18**, 231–254.
5. Blum, A., and Chalasani, P. (1993), An on-line algorithm for improving performance in navigation, *in* "Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci. (FOCS 93)," pp. 2–11.
6. Blum, A., Raghavan, P., and Schieber, B. (1997), Navigating in unfamiliar geometric terrain, *SIAM J. Comput.* **26**, 110–137.
7. Dudek, G., Romanik, K., and Whitesides, S. (1998), Global localization: Localizing a robot with minimum travel, *SIAM J. Comput.* **27**, 583–604.
8. Fleischer, R., and Trippen, G. (2000), Optimal robot localization in trees, *in* "Proc. 16th Annu. ACM Sympos. Comput. Geom.," pp. 373–374. A video shown at the 9th Annual Video Review of Computational Geometry.
9. Guibas, L., Motwani, R., and Raghavan, P. (1992), The robot localization problem in two dimensions, *in* "Proc. 3rd ACM-SIAM Symp. on Discrete Algorithms."
10. Kalyanasundaram, B., and Pruhs, K. (1993), A competitive analysis of algorithms for searching unknown scenes, *Comput. Geom. Theory Appl.* **3**, 139–155.
11. Kleinberg, J. M. (1994), The localization problem for mobile robots, *in* "Proc. 35th IEEE Symp. on Foundations of Computer Science," pp. 521–531.
12. Mehlhorn, K., and Näher, S. (1995), LEDA: A platform for combinatorial and geometric computing, *Commun. Assoc. Comput. Mach.* **38**, 96–102.
13. Miller, D., Atkinson, D., Wilcox, B., and Mishkin, A. (1989), Autonomous navigation and control of a Mars rover, *in* "Automatic Control in Aerospace" (T. Nishimura, Ed.), pp. 111–114, Pergamon Press, Oxford.
14. Papadimitriou, C. H., and Yannakakis, M. (1991), Shortest paths without a map, *Theorer. Comput. Sci.* **84**, 127–150.
15. Schuierer, S. (1997), Efficient robot self-localization in simple polygons, *in* "Intelligent Robots—Sensing, Modelling and Planning" (R. C. Bolles, H. Bunke, and H. Noltemeier, Eds.), pp. 129–148, World Scientific, Singapore.

16. Shen, C., and Nagy, G. (1989), Autonomous navigation to provide long-distance surface traverses for Mars rover sample return missions, *in* "Proc. IEEE International Symposium on Intelligent Control," pp. 362–367.
17. Sleator, D. D., and Tarjan, R. E. (1985), Amortized efficiency of list update and paging rules, *Commun. Assoc. Comput. Mach.* **28**, 202–208.
18. Thompson, W., Pick, H., Bennett, B., Heinrichs, M., Savitt, S., and Smith, K. (1990), Map-based localization: The 'drop-off' problem, *in* "Proc. DARPA Image Understanding Workshop," pp. 706–719.
19. Yacoob, Y., and Davis, L. (1988), "Computational Ground and Airborne Localization Over Rough Terrain," Technical Report CS-TR-2788, University of Maryland.