**Israel A. Wagner**

IBM Haifa Research Laboratory
Matam, Haifa 31905, Israel
wagner@il.ibm.com

**Michael Lindenbaum**
**Alfred M. Bruckstein**

Department of Computer Science
Technion City, Haifa 32000, Israel
mic@cs.technion.ac.il
freddy@cs.technion.ac.il

# MAC versus PC: Determinism and Randomness as Complementary Approaches to Robotic Exploration of Continuous Unknown Domains

## Abstract

*Three methods are described for exploring a continuous unknown planar region by a group of robots having limited sensors and no explicit communication. We formalize the problem, prove that its off-line version is NP-hard, and show a lower bound on the length of any solution. Then a deterministic* mark and cover *(MAC) algorithm is described for the on-line problem using short-lived navigational markers as a means of navigation and indirect communication. The convergence of the algorithm is proved, and its cover time is shown to be the asymptotically optimal* O(A/a), *where* A *is the total area and* a *is the area covered by the robot in a single step. The MAC algorithm is tested against an alternative randomized* probabilistic covering *(PC) method, which does not rely on sensors but is still able to cover an unknown region in an expected time that depends polynomially on the dimensions of the region. Both algorithms enable cooperation of several robots to achieve faster coverage. Finally, we show that the two methods can be combined to yield a third, hybrid algorithm with a better trade-off between performance and robustness.*

## 1. Introduction

Exploring unknown terrain is an important issue in robotics. The problem has already been investigated, and several methods have been suggested and implemented. Most of those methods, however, rely on complex, expensive, and fragile systems of sensors (e.g., odometers, infra-red sensors, ultrasound radar, or GPS), and extensive computational resources to run sophisticated mapping algorithms. In this paper, we suggest a minimalist approach: we wish to achieve the goal of covering with a minimum of sensing and computing, even if some performance reduction is implied. Our first algorithm uses a trail-laying mechanism and trail-following sensors, while the second (randomized) one uses only collision-detection sensors. The third method is a hybrid of the first two, combined to achieve a better trade-off between performance and robustness.

Existing methods for graph search (e.g., BFS and DFS) cannot be directly used for our purpose, since no vertices or edges exist in our setting; a robot can move to arbitrary points on the continuum, while the BFS and DFS algorithms assume a discrete and finite set of possible locations. Yet the approach of "forward" and "backward" steps (used in DFS in a discrete manner) can and will be used as an inspiration to our first algorithm, in which the discretization is achieved on-line by the robot.

Our first method, the deterministic MAC (mark and cover) algorithm, is similar in spirit to the famous *depth-first search* (DFS) algorithm. It is a local rule of motion that attempts to discover an uncovered point around the current location. If such a point exists, the robot goes over there. Otherwise, it backtracks to its previous location, using the marked trail. We prove that this method guarantees a complete covering of a connected region $R$ in no more than $2\lceil\frac{A+rP}{r^2}\rceil + 2$ units of time, where $A$ is the area of $R$, $P$ is its perimeter, and $r$ is the covering radius of the (circular) robot. In most real-life cases, $rP << A$, and the cover time amounts to $O(A/a)$, where $a = \pi r^2$ is the area covered by a robot of radius $r$ in a single step.

The International Journal of Robotics Research
Vol. 19, No. 1, January 2000, pp. 12-31,

Unfortunately, fully deterministic algorithms cannot be a complete answer for realistic robotic problems, since both sensors and effectors are extremely vulnerable to noise and failures. As opposed to purely computational problems, the environment of the robot is not known in advance, and even if it were, it may change during operation. Another limitation follows from the use of short-lived marks. The lifetime of the marked trail severely limits the amount of area that can be covered in a single shot of the deterministic MAC algorithm. One way to tackle these problems is to make the robot itself nondeterministic by introducing randomness into its behavior. This motivates our second algorithm for the covering problem. We call this second method PC—probabilistic covering. The basic rule of behavior here is to make a short step and then a random turn. Somewhat surprisingly, the expected performance of the PC approach is not so bad; it covers a convex region in average time $O(d(A/a)\log(A/a))$, where $d$ is the diameter of $R$.

A third approach is a combination of the first two. We show that a hybrid algorithm achieves an improved trade-off between the performance of the first method and the robustness of the second one.

Some related work has already been done in various areas.

### *1.1. Robotic Covering*

In previous work (Dudek et al. 1991; Deng and Mirzaian 1996), a discrete problem of graph exploration was solved using markers. More recently, the problem of covering a tiled floor was addressed in two different ways: in Wagner and Bruckstein's (1997) work, the dirt on the floor served as memory to help the robot's navigation, while in the works of Wagner, Lindenbaurm, and Bruckstein (1996, 1997), a vanishing trace was used for that purpose. Balch and Arkin (1994) addressed the issue of inter-robot communication in the context of various missions, among them *grazing*; i.e., visiting every point of a region for purposes of object-fetching. There, a reactive model of behavior was presented, and simulation showed that detailed communication does not contribute too much to the performance. Ciralt and Weisbin (1995) presented many experimental works for planetary exploration by autonomous robots. Heuristic navigation methods were given by Hofner and Schmidt (1995) for path planning of an autonomous mobile cleaning robot, and by Kuipers and Byun (1981) for a robot exploration and mapping strategy. However, no rigorous analysis was given in the above references. Hert, Tiwari, and Lumelsky (1996) presented an algorithm for exploration of an undersea terrain, using exact location sensors and internal mapping. Practical implementations of covering algorithms have been demonstrated by Yaguchi (1996) and Parker (1996). Yaguchi (1996) described a set of robots that help clean a railway station, using magnetic lines on the floor as guidelines. This method seemed to work well, but was limited to premapped regions. Parker (1996) created a cooperation of a team of robots by an explicit level of inter-robot communication. Each robot could choose one of multiple possible behaviors, according to its specific conditions. In one of these behaviors, the robot played the role of a janitorial service man by cleaning the dust around itself.

### *1.2. SLNMs: Short-Lived Navigational Marks*

The idea of using marked trails for searching is inspired by the Greek myth of Ariadne, the granddaughter of Zeus, who used a thread to escape from Dedalus's maze (Virgil 1971). One way to mark a trail is by using odor, like the pheromones used by various insects. Experiments with an insect-inspired robot were reported by Russell (1995), where the robot had an odor-marking and -detection system. Another way of marking is by heating the floor. According to several experiments reported by Borenstein, Everett, and Feng (1996), the temperature distribution $T$ at a distance $\delta$ from the trail and at a time $t$ after laying the trail can be approximated by

$$T(\delta, t) = I(t)e^{-(\delta/w)^2}, \tag{1}$$

where $I(t)$ is a time-variant intensity function of the thermal path, and $w$ is a constant. In a later work (Russell 1997), an experimental study was reported that showed how a thermal trail can be traced several tens of minutes after being marked. In some cases, leaving a thread on the floor or marking it otherwise, even temporarily, is not possible or is not sufficiently accurate. In such a case, we can replace the marking operation by similar "marks" in the robot's memory, where a map of the covered region is incrementally created. Such a device, however, would need very accurate (i.e., expensive) odometric or GPS sensors, as well as sophisticated methods of error correction, like the method described by Borenstein and Feng (1996) for correcting odometric errors by occasionally using an external positioning system. Other researchers (e.g., Bessiere et al. 1995) used robot-placed landmarks to help path planning in the presence of dynamic obstacles. In the current work, however, we will use traces so that our robots can be rather cheap and simple.

### *1.3. Randomization and Uncertainty in Robotic Tasks*

Uncertainty is an inherent factor in any real-life action, in particular, one that relies on the information gained from sensors and manipulations performed by actuators. One way to cope with uncertainty is *randomization*—introducing a random selection into the robot's control. In the works of Erdmann (1992) and La Valle and Hutchinson (1996), randomization was used to (partially) overcome uncertainty in various robotic tasks. In a sense, our PC algorithm is an extreme case of randomization, whereas no sensors are used. The case of randomized robotic covering (e.g.,the PC method discussed in our work) is different, however, because the probability of a point to be chosen as the next location of the robot depends on

both the previous location and the shape of the region being explored. Gage (1993) made a comparison between coordinated and random-search strategies, based on simulations and on a limited analysis of convex domains. In this paper, we attempt to continue the effort using a rigorous analysis of deterministic and random strategies.

### *1.4. Stochastic Coverage Processes*

The rate of coverage of graphs by random walk has been studied intensively (e.g., Aleliunas et al. 1979; Barnes and Feige 1993; Broder et al. 1994). Representative results in this context are the upper bounds of $O(mn)$ on the cover time of a graph with $m$ edges and $n$ vertices, and $O(mR \log n)$ where $R$ is the resistance of the graph, assuming all edges to be 1-Ohm resistors. On the other hand, coverage of continuous domains is less investigated. The most comprehensive text known to us is that of Hall (1988), where a coverage process is considered as a sequence of patches whose centers comprise a standard random process. This approach does not assume any correlation between the locations of consecutive patches; hence it is not directly applicable to robotics, where motion has a significant cost. We attempt to improve the situation in our PC approach, which is an extension of random walk from graphs to the continuum.

### *1.5. Locating a Robot*

The reason we suggest using external traces rather than memorizing the path in the robot's internal storage, is that the latter method requires a precise positioning system, which is hard to achieve. Neither odometric ("dead reckoning") or common GPS (Global Positioning System) methods are as yet sufficiently accurate for an indoor small-scale application. (See Borenstein and Feng's (1996) work for an analysis of the effects of odometric errors on motion planning, and a method of correcting them using an external positioning system.) In the work of Enge and colleagues (1996), the precision achieved by GPS so far is estimated by 8 m, which is far below the level of precision required for covering indoor regions by robots of radius 20–40 cm.

### *1.6. Off-Line Covering*

An off-line version of the problem as well as approximation algorithms for it are presented by Arkin and Hassin (1994). The related (NP-hard) problem of the optimal watchman route is to find the shortest path in a polygon such that every point of the polygon is visible from a point of the path. This problem is investigated by Chin and Ntafos (1986). The problem is also NP-hard where the watchman has limited visibility; such a problem is called "$d$-sweeper," and is discussed by Ntafos (1992).

### *1.7. A Decentralized Approach*

The problem we address is different from those addressed in other works in that we confine our robots to (at most) local sensing, such that the group is a decentralized one, and adding or deleting robots does not introduce a need to change the protocol. Also we prove the convergence of our algorithms and evaluate their performance by upper bounds on the cover time.

The rest of the paper is organized as follows. In Section 2 we define the basic problems and terms, prove the hardness of the off-line covering problem, and show a lower bound on the time required by any solution. In Section 3 we describe the MAC algorithm, prove its convergence, and analyze its performance. Then, in Section 4 we describe and analyze the PC algorithm for convex and nonconvex regions. Section 5 is devoted to the hybrid MAC + PC approach, and in Section 6 we show simulations of the three algorithms. Section 7 concludes with a discussion and some open questions.

## 2. Preliminaries

In this section, we define some basic terms to be used throughout the paper, show that our basic problem is NP-hard, and prove a lower bound on the time of coverage for any covering algorithm.

A rule of motion is *local* if it relies on the information available in the robot's near neighborhood. Our problem is to find a local rule of motion that will cause the robot to follow a *space-covering curve*, such that every point of the given region should be in some prespecified $r$-neighborhood of the robot's trail. Such a rule, if obeyed for $T$ units of time, should lead the robot to follow a piecewise polygonal curve $z(t)$ defined by the points $z(1), z(2), \ldots, z(T)$, that *covers* a connected planar region $R$; i.e.,

$$R = \bigcup_{t=0}^{T} B_r(z(t)), \tag{2}$$

where $B_r(z)$ is the intersection of $R$ and a disk of radius $r$ that is visible from $z$.

A visible circle is defined in the following way:

DEFINITION 1. If $z = (x, y)$ is a point and $R$ is a region in the Euclidean plane, then a visible $r$-circle around $z$ is defined as

$$C_r(z) \triangleq \left\{ z' \text{ such that } L(z, z') \in R \quad \text{and} \quad \left| z - z' \right| = r \right\},$$

where $L(z, z')$ is the straight line between $z$ and $z'$. In other words, this refers to those points on the circle that are visible from $z$; we do not assume seeing through obstacles or boundaries. Similarly,

DEFINITION 2. A visible open $r$-disk (or $r$-neighborhood) around $z$ in region $R$ is

$$B_r(z) \triangleq \left\{ z' \text{ such that } L(z, z') \in R \quad \text{and} \quad \left| z - z' \right| < r \right\}.$$

DEFINITION 3. An $r$-step path in $R$ is a sequence of points $Z = z_1, z_2, \ldots, z_m$ such that

$$\text{for all } i,\ 1 \le i < m : L(z_i, z_{i+1}) \in R \quad \text{and} \quad |z_i - z_{i+1}| = r.$$

DEFINITION 4. A set $R$ is $r$-path-connected if for any two points $z_s, z_d \in R$ there exists a finite $r$-step path $z_s = z_1, z_2, \ldots, z_m$ such that $z_d \in B_r(z_m)$. In other words, each point in $R$ can be covered by some $r$-step path started from any other point in $R$.

Note that a space-covering path (like the one we are looking for) is related to two famous curves:

- The *space-filling curve* (Sagan 1994) is a continuous one-dimensional curve that fills a two-dimensional space. If $R$ is the 2-D space to be covered, then each and every point of $R$ should be on the curve. Such a curve is an $r$-step-covering path with $r \to 0$.
- On the other extreme of this spectrum, we have the *optimum watchman curve*(Chin and Ntafos 1986), where each point of a polygonal region $R$ should be visible by some point on the curve. This is a space-covering curve, but this time with a variable step size and with covering radius $r \to \infty$.

Let us now define the problem we are going to deal with in this paper:

DEFINITION 5. A step-covering (SC) problem:
**Given:** a real number $r$ and an $r$-path-connected set $R$.
**Find:** a sequence of points $z_1, z_2, \ldots, z_T$ that are an $r$-step path in $R$ such that

$$R = \bigcup_{i=1}^{T} B_r(z_i).$$

Such a problem is referred to as off-line if the set $R$ is known in advance, and as on-line if at time $t$, the only known part of $R$ is the neighborhood of the path traversed so far.

Our algorithms are *on-line* methods; i.e., they decide on the next step without having the full details of the problem, which in our case could be a complete map of the region. Clearly, the on-line version of a problem is at least as hard as the off-line one; hence, we first discuss the hardness of the off-line SC problem, formally defined as follows:
**Given:** a real number $r$, an integer $m$, and an $r$-path-connected set $R$ (not necessarily simple).
**Question:** is there a step path of length $m$ that $r$-covers $R$? Namely, is there a sequence of points $z_t,\ t = 1, 2, \ldots, m$ such that

1. The sequence $z_t$ is an $r$-step path in $R$; i.e.,

$$\forall t \text{ s.t. } 1 \le t < m : L(z_t, z_{t+1}) \in R$$
$$\text{and } |z_{t+1} - z_t| = r,$$

2. The region $R$ is covered by the path, i.e.,

$$\forall z \in R, \quad \exists t_0 \in \{1, 2, \ldots, m\} \quad \text{such that } z \in B_r(z_{t_0}).$$

THEOREM 1. Off-line SC is NP-hard.

**Proof.** We shall show that our problem is at least as hard as the problem of finding a Hamiltonian path[1] in a grid graph, which is known to be NP-complete (Itai, Papadimitrious, and Szwarefiter 1982). This purpose will be obtained by showing that if one knows how to solve any instance of the SC problem in polynomial time, then any given instance of the Hamiltonian path problem can also be solved in polynomial time. Assume that we get an instance of the Hamiltonian grid-graph problem, made of $G(V, E)$ where the vertices in $V$ are grid points and the edges in $E$ are grid lines, and we are asked to find whether there is a Hamiltonian path in $G$. We build an instance for SC by embedding the given graph on a grid of size 1, and drawing a cross with edges of length 0.5 centered on each vertex. The region $R$ of the SC instance will be the union of all the crosses (see Fig. 1).

We now claim that there is a Hamiltonian path in $G$ if and only if there is a 0.5-covering path of $R$ with exactly $2n - 1$ points, where $n = |V|$. To see it, consider the following observations:

- If $G$ is Hamiltonian, then the Hamiltonian path is also a 0.5-covering path of $R$ with $2n - 1$ points, which are the $n$ original vertices plus $n - 1$ points in the middle of traversed edges— recall that the grid size is 1 while $r = 0.5$.
- Assume that there is a minimal 0.5-covering path of $R$, and denote it by the sequence $z_t,\ t = 1, 2, \ldots, 2n-1$. Now observe that a minimal covering should visit each vertex at least once; otherwise, it could be shortened. Hence, by removing the $n - 1$ nonvertex points from the sequence, one gets a Hamiltonian path in $G$. See an example in Figure 1. Hence, a path over $n$ points can cover a length of at most $2n$, with this maximum obtained only when the $n$ vertex points are visited. On the other hand, $2n$ is exactly the size of $R$; hence an $n$-long 0.5-covering path in $R$ corresponds to a Hamiltonian path in $G$. □

The theorem implies that finding an optimal covering path is a hard problem, (unless $P = NP$). In addition, we can

1. A *Hamiltonian path* is a sequence of vertices in a graph $G$ such that there is an edge between any two consecutive vertices, and each vertex of $G$ appears exactly once in the path.

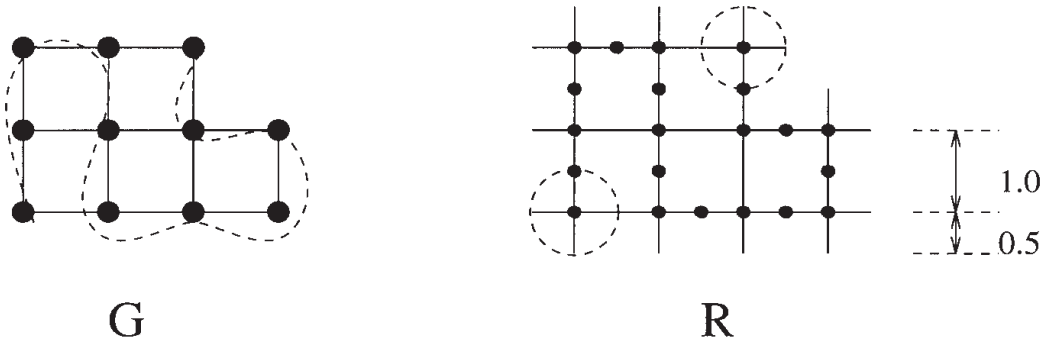


Fig. 1. $G$ (left), an instance of the Hamiltonian grid-graph problem, is transformed into $R$ (right), an instance of the step-covering problem with $r = 0.5$. The Hamiltonian path in $G$ (dashed line) corresponds to the 0.5-covering sequence in $R$ (dots), whose first and last covering circles are also sketched.

also show a lower bound on the length of any covering path, independent of the algorithm used to create the path.

LEMMA 1. The number of points in a covering sequence of $r$-circles, say $Z = z_1, z_2, \ldots, z_{T_c}$, such that $|z_{i+1} - z_i| \leq r$, is bounded from below

$$T_c \geq \left\lceil \frac{6\pi}{4\pi + 3\sqrt{3}}(A/a) - 1 \right\rceil,$$

where $A$ is the region's area and $a = \pi r^2$—the area covered by the robot in a single step.

**Proof.** In each step (except, perhaps, the first one), the robot jumps a distance of $r$, and hence (due to overlapping) adds at most $(\sqrt{3}/2 + 2\pi/3)r^2$ to the covered area. Thus, after $T$ points, the covered area is at most $S_T = (T-1)(\sqrt{3}/2 + 2\pi/3)r^2 + \pi r^2$. By equating $S_{T_c}$ to $A$, the lemma is implied. □

REMARK 1. It is intuitively reasonable to assume that as $r$ decreases, the "quality of covering" improves; i.e., the amount of overlap reduces. This intuition is made clear by the following result from Kershner (1939). Define $N(r)$ as the minimum number of $r$-circles needed to cover a region of area $A$. Then

$$\lim_{r \to 0} N(r) = (2\pi/\sqrt{27})(A/a), \tag{3}$$

and the minimum is attained in the "honeycomb" (hexagonal) arrangement of the circles, obtained by tiling the plane with congruent regular hexagons and circumscribing each hexagon with a circle. Note that the eq. (3) result from Kershner (1939) implies that, asymptotically, $T_c$ cannot go below $1.209\ldots \times (A/a)$, while Lemma 1 implies that for *any* value of $r$, $T_c \geq 1.06\ldots \times (A/a)$.

In many practical situations, the on-line version of the SC problem is more relevant than the off-line version, since an efficient on-line solution enables an autonomous robot to cover a region without the need to be preprogrammed with a detailed map. Other advantages of the on-line approach are the ability to react to changes in the geometry and topology of the environment, and the flexible mode of cooperation that can only be achieved via the on-line approach, while the preprogramming one is severely limited in this respect.

In this paper, we shall consider both a *deterministic* covering algorithm, in which the cover time is guaranteed, and a randomized one, in which we bound the expected cover time from above.

## 3. MAC (Mark and Cover): A Deterministic, Trail-Based Algorithm for the On-Line SC Problem

Consider a group of robots with very limited sensors that cooperate to cover a continuous, bounded, connected region. The robots have no means to calculate their locations or to communicate with each other directly. Their only means of communication are short-lived marks they draw on the floor (e.g., heat or smell), which enable them to recognize a location as "already visited." The intensity of the mark enables the robot to distinguish recently visited from previously visited points. The task of the robot(s) is to cover the whole region using the ability to follow a marked trail on the floor. If an odometric sensor of high precision is available, visited locations can be recorded in the robot's memory rather than marked on the floor. Our assumed robot covers a disk of radius $r$ around its center, and can move in an arbitrary direction either in a "forward" manner, where a new segment is marked down, or in a "backward" manner, at which time a previously marked segment is followed in a direction opposite to the direction of its marking. The robot can identify the direction of the trail by its intensity—if a point was marked later than another point, its mark will be stronger. Being at $z$, we assume the robot to be able to cover $B_r(z)$—an open disk of radius $r$ around $z$, and to sense (e.g., see or smell) the status of covering on $C_r(z)$— the boundary of the disk. For that purpose, the

robot should be able to sense trails as far as $2r$ away from $z$ to recognize a point on $C_r(z)$ as "uncovered." It is also assumed that the marked trails remain detectable during the execution of the algorithm, an assumption that will later be relaxed in a randomized version of the algorithm.

We now consider a reactive rule of motion such that a set of robots that obey this rule will eventually cover the region in the sense defined by eq. (2). The algorithm is called MAC for "mark and cover."

/* MAC—Mark and Cover */
/* $r$ = covering radius */
Rule MAC($z$: current location)
A) cover $B_r(z)$, the open $r$-disk around $z$;
B) if there is a point $z' \in C_r(z)$ such that $z'$ is yet uncovered
(i.e., there is no marked point in $B_r(z')$)
then
C) go to $z'$, while marking the line $z \to z'$;
/* break ties by a heuristic */
else
D) if there is a point $z' \in C_r(z)$ such that
the line $z' \to z$ is marked
/* line direction is detected by intensity, see eq. (1) */
then /* backtrack */
E) go to $z'$;
F) else STOP ($R$ is covered).
end MAC.

See Figure 2 for an example.

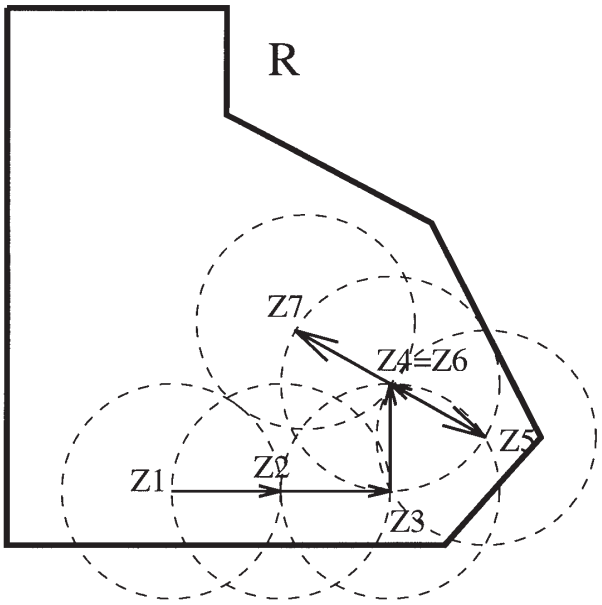


Fig. 2. Seven steps of the MAC procedure (bold arrows), using the "stay straight" heuristic to resolve ties. Note the step back from $Z_5$ to $Z_4$, due to the total coverage of the $r$-circle around $Z_5$.

The idea behind this algorithm is similar to a depth-first search (DFS): in a "forward" motion (steps B and C), the robot finds a new, uncovered area and covers it. In the process, it marks the path from the previous point to the new point. The direction of this mark can later be recognized, since if the line $z \to z'$ has been marked, the intensity at $z'$ will always be stronger than at $z$. Once no new point exists in the near neighborhood (i.e., on the circle $C_r(z)$), the robot "backtracks" using the marked path (step E), or, if no uncovered point nor backward mark exist around $z$, the robot finally stops (step F), being aware that its mission has been completed. Note that the robot should be able to sense trails as far as $2r$ away to recognize a point as "uncovered" in step B.

As for the heuristic in step C, we experimented with the following three heuristics for breaking the tie:

**Random.** Choose $z'$ at random from the uncovered points on $C_r(z)$.

**Peeling/Milling.** Choose the uncovered point $z'$ which is "rightmost" with respect to the line $(z_t, z_{t-1})$. Then, if the robot starts on a boundary point, the uncovered region is "peeled" from the outside inward, similar to the process of Wagner and Bruckstein (1997). Starting the same process from an internal (nonboundary) point, it will "grow" a clean area around it in a spiral process of milling.

**Stay Straight.** Keep going straight, as long as it is possible according to the MAC rule. This approach has the advantage of saving turns, thus reducing time and mechanical degradation in a realistic robotic setting.

Judging from our (limited) simulations, all three heuristics seem to have similar effects on the performance.

In the sequel, we show that the covering mission is indeed achieved by the algorithm, and we then analyze its timing performance.

### *3.1. Performance Analysis of the MAC Rule*

Assume that the robot has stopped at time $T$, and let us call the points where the robot has stopped, i.e., the sequence $z(1), z(2), \ldots, z(T)$, *vertices*, and the lines that were marked in one time or another *edges*. It is clear that the graph $G(V, E)$, where $V$ are the vertices and $E$ are the edges, is connected, since upon the creation of a new vertex $z'$ (step B), an edge is always being marked (step C) that connects $z'$ to the rest of the graph. Let us first show that each edge in $G$ is visited exactly twice during the execution of the MAC algorithm.

LEMMA 2. Under MAC, a marked segment $z \to z'$ is visited exactly once in each direction.

**Proof.** The segment $z \to z'$ may be visited either upon detecting $z'$ as a new point, at which time the mark is laid on

the segment, or in the course of backtracking from $z'$ to $z$. Once the segment is marked, its $r$ neighborhood is considered "marked" (by step B); hence, the only way to go over it again is by backtracking. But backtracking can happen once only, since after it happens once, the neighborhood is covered and won't be entered again. Hence it is the second (and last) traversal of this segment. □

Using the above result, we now proceed to show that the time of coverage can be bounded from above by a geometric function of $R$. To derive this result, we need an upper bound on the amount of area added to $R$ by extending it outward.

LEMMA 3. If a planar region $R$ is expanded by a strip of width $r$ to all directions (see Fig. 3), then its area increases by at most $r(P + \pi r)$, where $P$ is the total perimeter of $R$.

**Proof.** Let us first assume that $R$ is a simply connected domain, and its boundary is defined by $\alpha : [0, P] \rightarrow \mathbf{R}^2$; a planar, regular, simple, closed curve, parameterized by $\alpha$, the arc length along $\partial R$. Now consider a dwarf going along the boundary in the positive direction (i.e., with the inside to his left); an infinitesimal straight step is $ds$ long. Then the infinitesimal part of the expanded area is just a trapezoid with one base equal to $ds$ and the other base $ds(1 + rk(s))$, where $k(s)$ is the local curvature of $\alpha(s)$. See Figure 4.

The increment in the strip's area is at most $r(ds + rk(s)ds/2 + o((ds)^2))$, and hence $S$, the total area of the strip, is bounded as follows (terms dependent on $(ds)^2$ can be ignored as $ds \rightarrow 0$):

$$\begin{aligned} S &\leq \int_{s=0}^{P} r(1 + rk(s)/2)ds \\ &= r\left(\int_{s=0}^{P} ds + \frac{r}{2}\int_{s=0}^{P} k(s)ds\right) \qquad (4) \\ &= r(P + \pi r). \end{aligned}$$

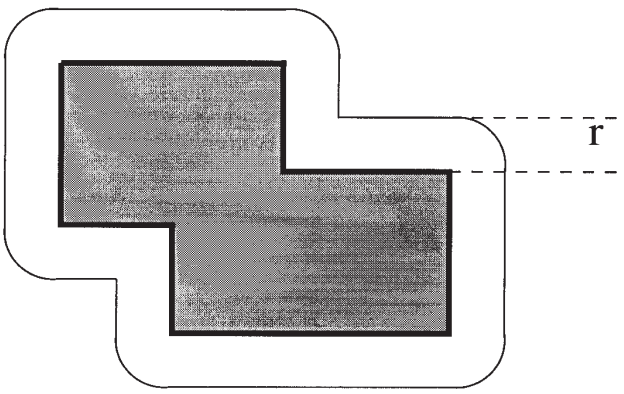


Fig. 3. An $r$-expanded shape.

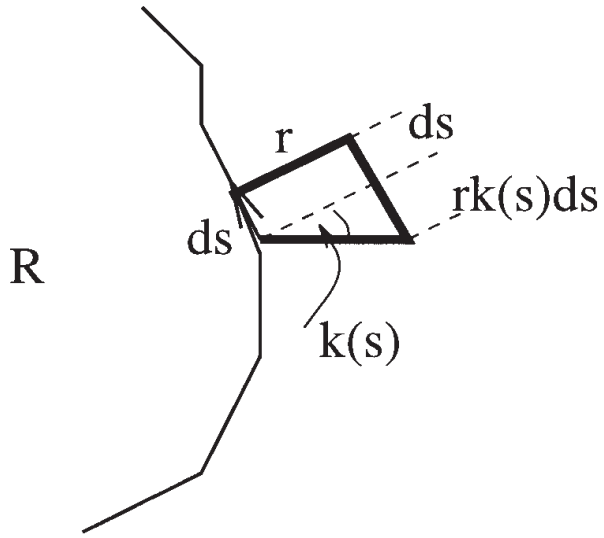


Fig. 4. An infinitesimal part of the $r$-wide strip around $R$ is approximated as a trapezoid (bold line) with area $r(ds + rk(s)ds/2 + o((ds)^2))$.

The last equality is obtained from the rotation index theorem.[2] Note that inequality (4) also holds for a nonsimple region if $P$ is the total perimeter, including the perimeters of the holes in $R$ if there are obstacles in it. □

REMARK 2. Inequality (4) becomes an equality if $R$ is convex, and then it is a case of the Steiner-Minkowski formula for polytopes (see, e.g., Berger 1987, p. 18).

Now we use Lemma 3 and a geometric observation on the MAC process to show an upper bound on the number of points visited in MAC.

LEMMA 4. If a region $R$ has area $A$ and perimeter $P$, then the graph $G(V, E)$ whose vertices are the visited points and whose edges are the lines marked by the MAC algorithm has at most $2\sqrt{3}\frac{A + rP + \pi r^2}{r^2}$ edges.

**Proof.** According to the algorithm, a point can never become a vertex if it is less than $r$ distance from any existing marked point. That is, if $(z, z')$ is an edge in $E$, then no third vertex of $V$ can be within $B_r(z) \bigcup B_r(z')$. (At this point we ignore the boundaries of $R$; this will be fixed in the sequel.) Hence, if we draw a rhombus made of two (30°, 120°, 30°) triangles around each edge in $E$, no two rhombuses will intersect; otherwise (if two rhombuses do intersect) it is implied that a third visited point exists in the $r$ vicinity of a vertex of the rhombus.

See Figure 5 for an example.

The area of such a unique quadrilateral region is $r^2/2\sqrt{3}$,

2. The rotation index theorem (see, e.g., Do-Carmo 1976, p. 396) says: If $\alpha : [0, P] \rightarrow R^2$ is a plane, regular, simple, closed curve, then $\int_0^P k(s)ds = 2\pi$, where $k(s)$ is the curvature of $\alpha(s)$ and the curve is traversed in the positive direction (i.e., with the inside to the left of the walker).

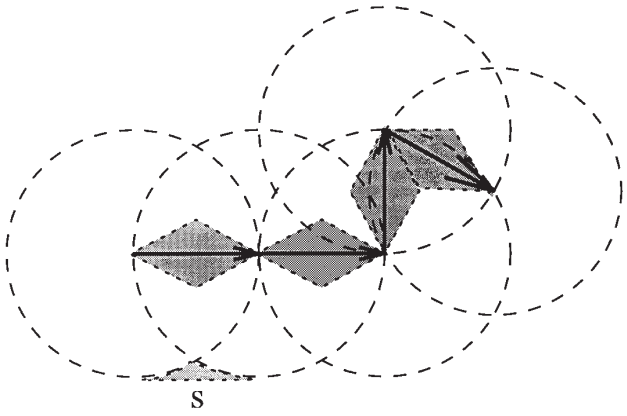


Fig. 5. Four steps of the MAC procedure (bold arrows), the covered area (dashed line), and the unique area around each edge (gray rhombus). Note that only the circles around the stopping points are covered, not those around the edges; for example, the area $s$ is as yet uncovered in this case. Also note that an angle between two edges in the path cannot be smaller than 60°; hence all rhombuses are disjoint.

so if we had a region with no boundary (e.g., a torus or a sphere), we would have that

$$|E| \leq \frac{2\sqrt{3}A}{r^2}.$$

However, in most realistic situations there are some additional edges near the boundary of $R$; the number of such edges can be bounded from above by assuming that the region $R$ has been expanded by a strip of width $r$. The area of such a strip does not exceed $S = r(P + \pi r)$, as is implied by Lemma 3 above. Hence, we get

$$|E| \leq 2\sqrt{3}\frac{A + rP + \pi r^2}{r^2}. \tag{5}$$

□

REMARK 3. The weight of the perimeter $P$ in eq. (5) may become significant; for example, if the area has the shape of an extremely acute triangle, the number of vertices may be arbitrarily large, although the area of $R$ is bounded. Another extremal example is a fractal shape (i.e., bounded area but unbounded perimeter) where the angles are not too small but the visibility is limited to a small area due to the infinite number of corners.

So far, we have shown that the cardinality of the edge set is bounded from above, and each edge is traversed twice; hence the algorithm should eventually stop. Let us now show that upon termination, the vertices of the graph are indeed a cover of the region $R$ in the sense of eq. (2).

LEMMA 5. The set $z_1, z_2, \ldots, z_T$ of vertices defined by the MAC algorithm covers the region $R$; i.e.,

$$R = \bigcup_{i=1}^{T} B_r(z_i).$$

**Proof.** Assume, on the contrary, that upon termination there still exist uncovered points in $R$. Consider $z_c$, the uncovered point that is closest to a point of the set $V$, say to $z_i$. Now the distance $|z_c - z_i|$ cannot be less than or equal to $r$, or otherwise no backtracking from $z_i$ was possible (i.e., $z_c$ should have been detected during one of the visits to $z_i$). Let us denote $|z_c - z_i| = r + \epsilon$ with $\epsilon > 0$; however only an *open* $r$ neighborhood of $z_i$ is covered. Hence, there must be an uncovered point in $R$ that is closer to the set $V$ than $z_c$ (e.g., a point on the line $L(z_i, z_c)$ that lies in distance $r + \frac{\epsilon}{2}$ from $z_i$), in contradiction to our assumption. □

Now, as the edge set of $G$ is bounded and each edge is traversed twice, one can show that $T$, the cover time, is bounded, too:

THEOREM 2. The time needed to cover a region $R$ with area $A$ and perimeter $P$ by the MAC algorithm, denoted $T^{\mathrm{MAC}}$, is bounded as follows:

$$2\left\lceil \frac{6\pi}{4\pi + 3\sqrt{3}}(A/a) - 1 \right\rceil \leq T^{\mathrm{MAC}} \leq 2\frac{A + rP + r^2}{r^2}.$$

**Proof.** The robot goes over an edge per unit of time. According to Lemma 2, each edge is traversed exactly twice, and from Lemma 4, we know that the number of edges is bounded above by $2\frac{A+rP+r^2}{r^2}$. Hence, the upper bound results. For the lower bound, we first apply Lemma 1 for a lower bound on the number of points (i.e., vertices in $G$) before covering, and then use the fact that the graph $G(V, E)$ is connected; hence there are at least $|V| - 1$ edges in $E$, each of which is traversed twice. □

REMARK 4. Our algorithm is "nearly optimal" in the sense that its covering time does not exceed a constant times the minimum possible covering time by any robotic cleaner with radius $r$, which is $\left\lceil \frac{6\pi}{4\pi+3\sqrt{3}}(A/a) - 1 \right\rceil$, as was shown in Lemma 1 above.

REMARK 5. Our analysis gives upper and lower bounds, but the exact time of covering depends on the number of vertices generated in the process, which in turn depends on the starting point(s), the shape of the region, and the heuristic used in step C of the algorithm to choose the next point among several uncovered points in $C_r(z)$. In the course of our simulations, we experimented with several heuristics for breaking the tie; namely, the "random," the "peeling/milling," and the "stay straight" approaches that were explained in the previous subsection. According to our simulations, there is no

great difference in cover time between the heuristics. Hence, we believe that in practice the best approach is the third one, which is more efficient due to the reduced number of turns.

REMARK 6. It is quite easy to see that the path obtained by MAC is not necessarily optimal (i.e., shortest). An optimal path may be found if the whole map is given; such a problem is an off-line problem (as shown above, even this problem may be hard to solve). However, as shown by Theorem 2, the on-line path achieved by MAC when only local data is available is not longer than a constant times the optimal path; hence, a MAC solution has a property known as a constant *competitive ratio*.

REMARK 7. The trace resulting from a run of the MAC algorithm can serve as a kind of roadmap to the unknown region. By counting steps, the robot(s) may put (at every node during the backtracking stage) the path length and maybe other information on the subtree, which may be useful for the next robots/people to come. In this way the robot contributes implicitly to a global plan, without being aware of anything beyond its own part.

### *3.2. Multi-Robot MAC: Collisions and Cooperative Covering*

There is a good potential for cooperation when several robots apply the MAC rule, and there is no need for additional hardware or communication protocols. As can be easily seen, collisions are not a problem under the MAC protocol (assuming the hardware is not vulnerable to bumps), since a robot can consider its fellow as a "wall." Still, some performance degradation is likely to occur if too many of the robots are cluttered in a small neighborhood. Also, there is no danger of deadlock, since a robot never "waits"—it either goes backward or forward, (or makes a final stop if neither kind of step is possible). Recall that according to the rule, a marked line never intersects either itself or another robot's thread; hence, several robots will perform at least as well as one. See Figures 6 and 7.

However, the exact degree of speedup is dependent on the geometry of the region and the initial location of robots. The basic problem is that if several robots are initially located very closely, one may put its thread around another and block it, thus reducing the speedup ratio. We suggest two possible ways to overcome the perpetual blocking problem:

**The deterministic method.** As an initial step, all robots should make a tour around the region and distribute equidistantly along the boundary. Then each robot should follow the MAC rule, and in case of a tie (i.e., several possible continuations to uncovered points), choose to go to the point that is closest to its starting point. This way, each robot first takes care of its neighborhood before interfering with others, thus postponing blockages to a later time.

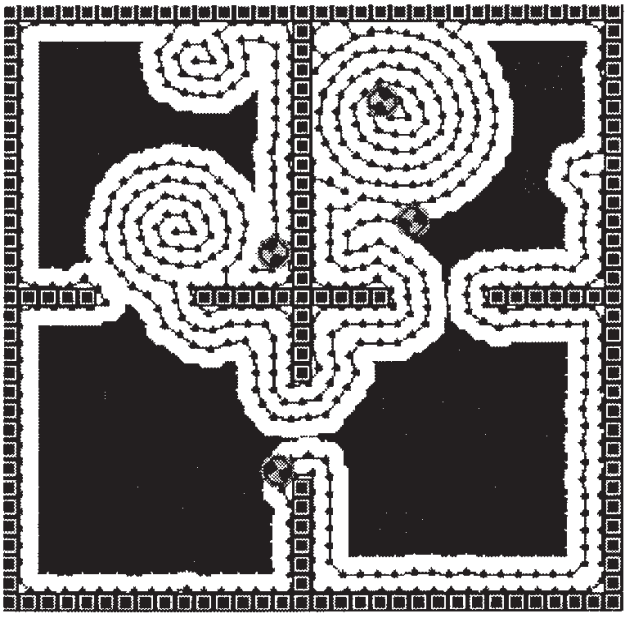

Fig. 6. Four robots using the MAC algorithm in four rooms: preliminary stage.

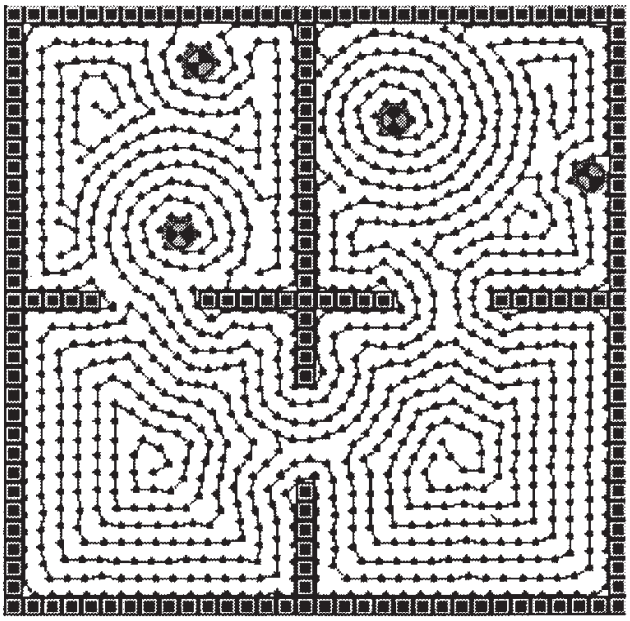

Fig. 7. Four robots using the MAC algorithm in four rooms: final stage. Note that their traces are four separate trees.

**The randomized method.** Once a blockage is created, a blocked robot, rather then going into "rest" state, can start walking around randomly until it finds an uncovered area. This kind of step, however, should be taken only if all area is not yet covered; for this awareness, we'll need some limited form of global communication.

### *3.3. Drawbacks of MAC*

The MAC algorithm has two important advantages: it is efficient in time, and aware of completion. However, it has its drawbacks, such as its dependence on sensors and traces, which results in a sensitivity to noise in both sensors and the environment. Another shortcoming is that multi-robot cooperation is rather limited unless the geometry is known is advance. In the next section, we propose a complementary approach: an almost sensorless covering algorithm, which uses random decisions to guarantee coverage with high probability.

## 4. PC (Probabilistic Covering): A Randomized Approach to the Covering Problem

In this section, we consider a robot that acts without any sensory inputs; it makes a step, chooses a random new direction, and then makes another step. Clearly, the average performance of this method is lower than MAC, but it has the advantage of being sensorless, so it can serve as a complementary approach to the deterministic one. In the sequel we shall refer to the $r$ disk around $z$ by $B_r(z)$, and to the $r$ circle around $z$ by $C_r(z)$. We shall also denote by $\mu(z)$ the maximum $x$ such that $x \leq r$ and $B_{2x}(z) \in R$. Actually, calculating $\mu(z)$ is the only place where we need sensors in the system. Formally, the rule of motion is defined as follows:

/* PC - Probabilistic Covering */
/* $r$ = covering radius */
Rule PC($z$: current location)
A) cover $B_r(z)$;
B) set $\mu(z) = \min\left\{r, \max_{(B_{2r'}(z)\subset R)}\left\{r'\right\}\right\}$;
/* $\mu(z)$ is half the maximum radius */
/* (not exceeding $r$) */
/* of a circle around $z$ within $R$ */
C) choose a random neighbor $w$ from $C_{\mu(z)}(z)$;
D) go to $w$;
end PC.

See Figures 8, 9, and 10 for examples of the process.[3] As opposed to the previous algorithm, PC has no stopping criterion. In practice, one can use the analytical bound on the expected cover time (to be derived in the sequel) as an estimation of the time at which the robot can be stopped. Note that if $C_r(z)$ intersects the boundary of $R$, then the duration

3. A JAVA simulator of the PC process is available on the web at http://www.cs.technion.ac.il/~ wagner/pub/mac.html.

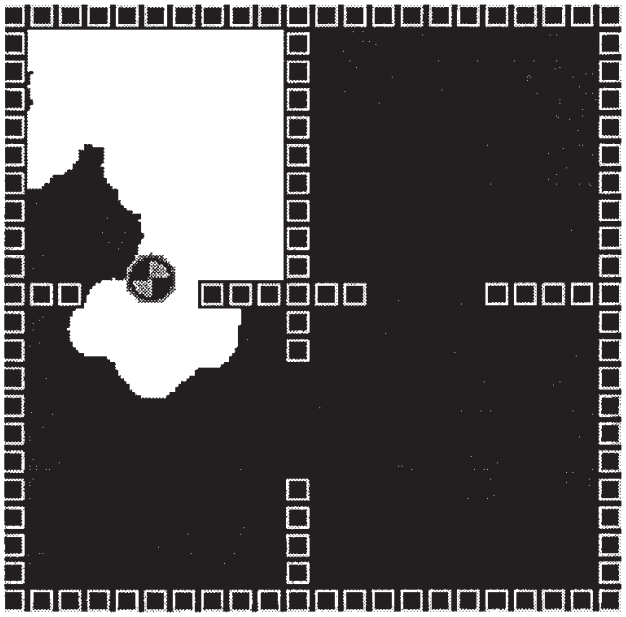

Fig. 8. A lonesome PC robot; the gray area has not yet been covered.

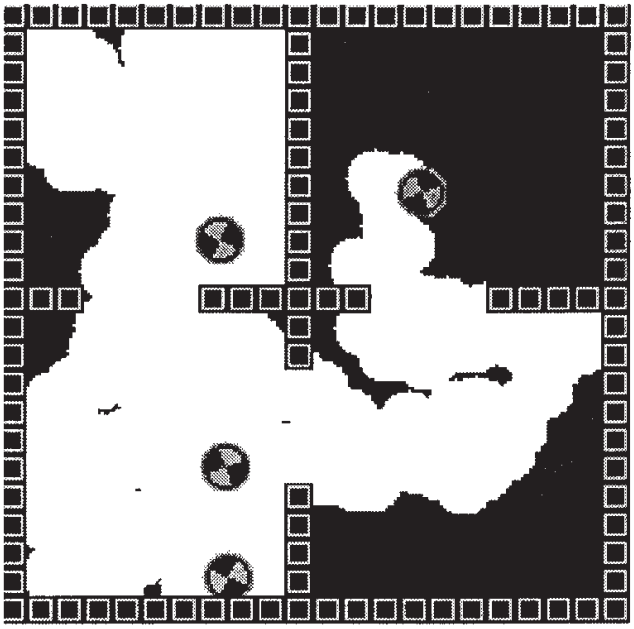

Fig. 9. Four PC robots working together. A fellow robot is considered as an obstacle; hence no collisions should occur according to the PC rule.

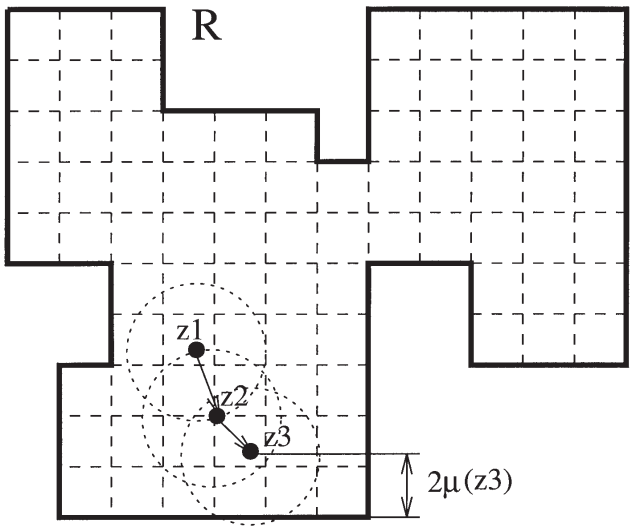


Fig. 10. A grid-polygon $R$, partitioned into unit squares, and a possible sequence of PC steps that take random continuous locations $z_1, z_2, z_3$, thus covering the dashed circles. In this case, $\mu(z_2) > \mu(z_3)$, and hence the step size at time $t = 2$ is greater than at time $t = 3$. The dashed circles designate the covered area. Note that since the covering radius is always $2^{1/2}$ while the grid size is 1, it is sufficient to visit all squares to guarantee a coverage of $R$.

of a PC step shall be shorter than one unit of time, since the step length is $\mu(z) < r$. In each step, the robot scans around to see if a boundary exists within distance $r$; hence, we shall assume that the time spent at $z$ is proportional to $(\mu(z))^2$, where $\mu(z)$ is half the maximum radius not exceeding $r$ of a circle around $z$ within $R$. Thus, the time spent in traversing the sequence of $k$ points $z_1, z_2, \ldots, z_k$ is equal to $\sum_{i=1}^{k} \mu^2(z_i)$, where $z_{i+1} = z_i + \mu(z_i)e^{j\theta_i}$, $\theta_i$ being the angle chosen at the $i$th step. The reason for making the step length half the possible maximum is to avoid the chance of the robot going to $\partial R$, where it will get stuck forever since $\mu(z)$ (i.e., the step size) vanishes on the boundary.

We model the robot as a point that covers a circle of radius $r$ around itself. Because of the random nature of PC, no deterministic bound can be stated on the cover time; we shall, however, draw some bounds on the *expected* cover time and its *variance*, and both will be given as functions of the electrical resistance of a conductive material in the shape of $R$. This resistance can be further related to the geometrical properties of the robot and the region. More specifically, we prove the following:

**Expected time of complete coverage $\mathbf{E}\left[T^{\mathrm{PC}}\right]$.** The expected time until full coverage of $R$ (a unit-grid polygon of size $n$ by a PC robot that covers a radius of $\sqrt{2}$ in each step), is bounded by

$$n\rho \leq \mathbf{E}\left[T^{\mathrm{PC}}\right] \leq 2n\rho \log n,$$

where $\rho$ is the electrical resistance of $R$ (assuming a material of unit *sheet resistance*, to be defined in the sequel). Note that the resistance $\rho = \rho(R)$ can sometimes be bounded in terms of the geometrical properties of the shape, and can always be numerically approximated. For example, if $R$ is a $\sqrt{n} \times \sqrt{n}$ square, then its resistance is $O(\log n)$, when measured between the bottom-left and a top-right squares. In case of an $a \times b$ rectangle with $a << b$, $\rho = O(b/a)$. Recall from Lemma 1 that *any* covering path should have at least $\left\lceil \frac{6\pi}{4\pi+3\sqrt{3}}(A/a) - 1 \right\rceil$ steps.

**Variance in the Cover Time.** The variance in time of complete coverage, $\mathbf{V}\left[T^{\mathrm{PC}}\right]$, is bounded from above:

$$\mathbf{V}\left[T^{\mathrm{PC}}\right] \leq 2^{11}n\rho,$$

which yields an upper bound on the standard deviation of the cover time:

$$\sigma\left[T^{\mathrm{PC}}\right] = \sqrt{\mathbf{V}\left[T^{\mathrm{PC}}\right]} \leq 32\sqrt{2n\rho}.$$

Our results can be extended to more general shapes, but this involves various types of cumbersome details that will be omitted in this paper. Note that the above results are achieved without using any sensors except collision detectors (the robot cannot distinguish "tiles" or "grid squares"), and thus have almost no vulnerability to noise. They can be used as is, or combined with a sensor-based algorithm to achieve a trade-off between cover time and coverage guarantee.

### *4.1. Analysis of the Cover Time by PC*

There are a wealth of results in the literature for cover times by random walk on graphs, a sample of which was mentioned in the introduction. Our case is different, however, since the robot can be in any point in the continuum of the region, rather than being bounded to a finite set of such points. One may wish to partition the region into squares, considering a random walk on a graph with the set of squares as its vertex set; however, this will not do, because the transition probabilities are not constant; rather, they depend on the precise location of the robot within a square. [4] Hence, we shall use continuous arguments to analyze the process.

We first observe that the PC process is a strong Markov process, since the probability of visiting a location in the next step depends only on the current location but not on earlier history—the robot has no memory at all. For such a process, it was proved by Matthews (1989) that if $Q = \{q_1, q_2, \ldots, q_n\}$ is a collection of subsets of a set $R$, then $\mathbf{E}[T(q_1, q_2, \ldots, q_n)]$, the expected time for visiting some

4. In other words, the process is not *time homogeneous.*

point of every subset in $Q$ (starting from anywhere in $R$) is bounded as follows:

$$h_{\max} \leq \mathbf{E}\left[T(q_1, q_2, \ldots, q_n)\right] \leq h_{\max} \sum_{i=1}^{n} 1/i, \qquad (6)$$

where

$$h_{\max} = \max_{x \in (R \setminus Q), 1 \leq i \leq n} \{h_i(x)\},$$

and $h_i(x)$ is the expected time to first reach subset $q_i$ upon starting from $x \in R$. Let us now assume that $R$ is partitioned into a set of unit squares, $S = \{s_1, s_2, \ldots, s_n\}$. This partition is not known to the robot, but will serve us in our analysis. To establish bounds on the average cover time of the PC process, we further observe that (since the robot's covering radius is $r = \sqrt{2}$) if the robot has visited all the $n$ squares in $R$, then $R$ is totally covered. See Figure 10 for an example.

Clearly, if a robot is located anywhere within such a square, the whole square is covered (actually, some of the neighbor squares are also partially covered, but this does not harm our upper-bound result). Thus, visiting all the small squares is sufficient to guarantee a full coverage of $R$. On the other hand, to cover $R$ starting from any point in it, the robot should make, at least once, the tour between the two farthest squares in $R$. Let us define the *hitting time* (also known as the *access time* or *first-passage time*) from a point $x \in R$ to a square $s_j$, denoted $h_j(z)$, as the expected time of a PC process that starts at $z$ and ends upon first reaching a point in square $s_j$. We also define $C(x, y)$, the *effective commute time* between points $x$ and $y$, as the sum $h_{s(x)}(y) + h_{s(y)}(x)$, where $s(z)$ stands for the unit square that includes $z$. It is thus implied by eq. (6) (using $\sum_{i=1}^{n}(1/i) < 2\log n$) and the above observations that the expected cover time of $R$ can be bounded:

$$\frac{1}{2} \max_{x,y \in R} \{C(x, y)\} \leq \mathbf{E}\left[T^P C\right] \leq 2(\log n) \max_{x,y \in R} \{C(x, y)\}. \qquad (7)$$

To find the maximum effective commute time $C(x, y)$ in $R$, we now show that this time is proportional to the product of the number of squares in $R$ and the electrical resistance between $x$ and $y$, to be defined in the sequel. The following lemma is, in a sense, a continuous analog to the work of Chandra and colleagues (1997), which related the hitting time of a random walk on a graph with its electrical resistance, considering each edge as a 1-Ohm resistor.

LEMMA 6. $C(x, y)$, the effective commute time between points $x$ and $y$ in $R$, obeys the following equation:

$$C(x, y) = n\rho(x, y),$$

where $n$ is the area of $R$ and $\rho(x, y)$ is the electrical resistance between square points $x$ and $y$, assuming $R$ to be made of a uniform material with unit sheet resistance.[5]

**Proof.** Let us denote the maximum step size by $r$. In a step, the PC robot selects a random angle and goes in that direction. The length of the step is $\mu(z)$, half the maximum radius not exceeding $r$ of a circle around $z$ within $R$. As explained before, we assume the time spent at $z$ to be $(\mu(z)/r)^2$, which is one unit in an internal point of $R$ (i.e., where $\mu(z) = r$), and less near the boundary, where $\mu(z) < r$ and steps are shorter (see Fig. 10). If $z \notin s_j$, then the expected time to reach square $s_j$ from $z$ is just the average of the step length plus the access time over a $\mu(z)$ circle around $z$; i.e.,

$$h_j(z) = (\mu(z)/r)^2 + \frac{1}{2\pi} \int_{\theta=0}^{2\pi} h_j(z + \mu(z)e^{i\theta}) d\theta, \qquad (8)$$

where $z + \mu(z)e^{i\theta}$ refers to a point at distance $\mu(z)$ from $z$ and angle $\theta$ to the $x$ axis, in the complex notation. Clearly, if $z \in s_j$, then $h_j(z) = 0$.

Now consider $R$ as a flat surface of a uniformly resistive material with unit sheet resistance, and assume that a current of $I_0 = 4/r^2$ Amperes (A) per unit of area is uniformly injected into $R$, and $4n/r^2$ A are taken from $R$ via the square $s_j$. Let us also denote the electric potential at point $z$ relative to square $s_j$ by $\phi_j(z)$. Since there are no current sources within $R$, we know from the divergence theorem (see, e.g., Kaplan 1984, p. 319) that the amount of current entering the boundary of $R$ should equal the current exiting through it (i.e., the total current through the boundary should vanish). Due to symmetry and uniformity of the resistance, the average potential around a circle of radius $\mu$ can be calculated:

PROPOSITION 1. The average potential difference between the center and the circumference of a circle of radius $\mu$ on a uniform surface with unit sheet resistance, into which $I_0$ A of current are uniformly injected per unit area, is

$$\overline{\phi(\mu) - \phi(0)} \triangleq \frac{1}{2\pi} \int_{\theta=0}^{2\pi} (\phi(\mu e^{i\theta}) - \phi(0)) d\theta = \frac{I_0 \mu^2}{4}.$$

The proof of Proposition 1 is deferred to the Appendix. Choosing $I_0 = 4/r^2$, one gets $\overline{\phi(\mu) - \phi(0)} = (\mu/r)^2$, and hence (writing $\mu$ for $\mu(z)$ and $\phi_j(z)$ for the potential at $z$ when the potential in square $s_j$ is kept at zero):

$$\frac{1}{2\pi} \int_{\theta=0}^{2\pi} (\phi_j(z) - \phi_j(z + \mu e^{i\theta})) d\theta = (\mu/r)^2,$$

5. The *sheet resistance* of a material is defined as the voltage across a square of the material caused by one unit of current (i.e., 1 Ampere) that is flowing between two parallel edges of the square. The sheet resistance is commonly expressed in units of Ohms per square.

or

$$\phi_j(z) = (\mu/r)^2 + \frac{1}{2\pi}\int_{\theta=0}^{2\pi} \phi_j(z + \mu e^{i\theta})d\theta. \qquad (9)$$

From the equivalence of eqs. (8) and (9), and the uniqueness[6] of the expectation function $h_j(z)$, we see that $h_j(z)$ is equal to the potential $\phi_j(z)$ if $4r^{-2}$ units of current are injected into each unit of area, and $4nr^{-2}$ units of current are taken from square $s_j$, which is held in potential 0. In a similar way, one can show that $h_i(z) = \phi_i(z)$, if $4/r^2$ units of current are injected into each unit of area, and $4n/r^2$ units of current are rejected from $s_i$. Now if we reverse the direction of all currents in the second case, we find that $h_i(z) = -\phi_i(z)$, if $4/r^2$ units of current are taken from each unit of area across $R$, and $4n/r^2$ units of current are injected into $s_i$. Due to linearity of resistive electrical systems, we can superpose both sheets together, thus making all currents cancel each other, except the $4n/r^2$ A injected at $s_i$ and taken from $s_j$. This, together with Ohm's law,[7] implies that $C(x, y)$ (the effective commute time between points $x$ and $y$) obeys

$$\begin{aligned} C(x, y) &= h_{s(y)}(x) + h_{s(x)}(y) \\ &= \phi_{s(y)}(x) - \phi_{s(x)}(y) \\ &= \frac{4n}{r^2}\rho(x, y), \end{aligned} \qquad (10)$$

where $s(z)$ is the square including $z$ and $\rho(x, y)$ is the electrical resistance between points $x$ and $y$ in $R$. This resistance is measured as the potential difference between $x$ and $y$ while injecting a 1-A current into one square, say $s(x)$, and taking it from $s(y)$.

Substituting $r = \sqrt{2}$ in eq. (10) and considering the fact that superposing the two sheets implies half the original resistance yields the lemma. □

We now combine the above results to obtain the following theorem.

THEOREM 3.

$$n\rho \le \mathbf{E}\left[T^{\mathrm{PC}}\right] \le 2n\rho \log n,$$

where $n$ is the size of $R$ and $\rho$, its maximum resistance defined as $\rho = \max_{x,y\in R}\{\rho(x, y)\}$.

**Proof.** Immediate, by taking maximum on Lemma 6 and substituting in eq. (7). □

A corollary is implied for a square room:

COROLLARY 1. If $R$ is a square $a \times a$ room, then

$$c_1 a^2 \log a \le \mathbf{E}\left[T^{\mathrm{PC}}\right] \le c_2 a^2 \log^2 a,$$

where $c_1$ and $c_2$ are small constants.

**Proof.** (Sketch) We use the fact that the resistance of a square is $\Theta(\log a)$.[8] Then we also note that for an $a \times a$ room, $n = a^2$, which, substituted into Theorem 3, implies the corollary. □

Note that the upper bound on the expected cover time in this case is only $\log^2 a$ times the optimal possible $O(a^2)$.

### 4.2. An Upper Bound on the Variance of $T^{\mathrm{PC}}$

For our results to be useful, we now show that the variance of the cover time, denoted $\mathbf{V}\left[T^{\mathrm{PC}}\right]$, is also bounded from above and hence there is only a limited spread of the covering time around its average. It has been proved by Aldous (1991) that the variance in the cover time of a set $S$ is at most constant times the expected time of covering the last item in the set:

$$\mathbf{V}\left[T^{\text{cover of } S}\right] \le c_0 \cdot \mathbf{E}\left[T^{\text{cover of the last item in } S}\right],$$

where $c_o$ is a constant[9] less than $2^{10}$. Applying it to our case, we can use the maximum access time as an upper bound to the cover time of the last item (i.e., a yet-unvisited square), so we get

$$\mathbf{V}\left[T^{\mathrm{PC}}\right] \le 2^{10} \max_{x,y\in R}\{C(x, y)\} \le 2^{11} n\rho,$$

which implies that the standard deviation is at most $32\sqrt{2n\rho}$.

### 4.3. A Universal Traversal Sequence of Angles

Interestingly, we can prove the existence of a deterministic (derandomized) version of the PC algorithm that is also sensorless and guarantees covering within a polynomial number

6. The function $h_j(z)$ is uniquely determined by

$$\begin{aligned} h_j(z) &= \int_{t=0}^{\infty} t \cdot \mathrm{Prob}\left\{\text{square } s_j \text{ is reached from } z \text{ in a time in } (t, t+dt)\right\} dt \\ &= \int_{t=0}^{\infty} t \cdot \left[\sum_{k=0}^{\infty} \frac{1}{(2\pi)^k} \int_{\theta_1=0}^{2\pi}\int_{\theta_2=0}^{2\pi}\cdots\int_{\theta_k=0}^{2\pi} \Delta_t(\theta_1, \theta_2, \ldots, \theta_k)d\theta_1 d\theta_2 \ldots d\theta_k\right] dt, \end{aligned}$$

where $\Delta_t(\theta_1, \theta_2, \ldots, \theta_k) = 1$ if the sequence of angles $\theta_1, \theta_2, \ldots, \theta_k$ leads from point $z$ to (some point of) square $s_j$ in a time in $(t, t + dt)$, and 0 otherwise.

7. Ohm's law says that the voltage drop between two points is equal to the product of the current flowing between the points and the point-to-point resistance.

8. It is of interest to mention a lumped circuit analogy: a square $m \times m$ mesh of 1-Ohm resistors is known (Chandra et al. 1997) to have resistance $\Theta(\log n)$.

9. This value of the constant does not appear in Aldous's (1991) work, but can be calculated based on the analysis done there.

of time units. This process, in our opinion, is mostly of theoretical value, as practically, one would like to use the sensorless PC or the sensor-based MAC algorithm, which achieve much better times. Although the process described in this subsection is less efficient than the two previous algorithms, it still seems to carry a theoretical insight as an interesting case of derandomization.

Let us define a universal traversal sequence of angles (UTSA) for a family of planar sets $\mathcal{F}$ as a sequence of real numbers $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_t, \ldots$, all in $[0, 2\pi)$, such that if a PC robot takes the turn $\alpha_t$ in step $t$, it is guaranteed to cover any shape from $\mathcal{F}$, independent of the starting point. In other words, the robot starts at an arbitrary point $z_1$ and follows the rule $z_{i+1} = z_i + \mu(z_i)e^{j\alpha_i}$, $\alpha_i$ being the $i$th angle in the UTSA. In this section, we shall show that if $\mathcal{F}$ is the set of all $n$-size unit-grid polygons, (i.e., polygons made of $n$ attached $1 \times 1$ squares), then such a sequence exists and has a cover-time polynomial in $n$.[10] For this purpose, we follow the probabilistic method invented by Erdös and used by Aleliunas and colleagues (1979) to prove that a sequence of length $O(n^4 \log n)$ exists that covers any edge-labeled $k$-regular graph[11] with $n$ vertices.

THEOREM 4. There exists a sequence of angles that covers, within time $4n^4 \log n$, any rectilinear gridded polygon of size $n$.

**Proof.** First let us observe that if $\mathcal{F}$ is the set of all $n$-size unit-grid polygons, then $|\mathcal{F}| < 2^{n^2}$ (since all polygons of size $n$ can be enclosed by an $n \times n$ square). We next apply Theorem 3 to obtain an upper bound of $t = 2n^2 \log n$ on the expected cover time of any polygon in $\mathcal{F}$, using the fact that the resistance $\rho$ obeys $\rho \leq n$ for such polygons. Hence, after $2t$ units of time, the probability of complete coverage is at least $1/2$,[12] and after $mt$ units of time, it is at least $1 - 2^{-m}$. On the other hand,

$$\begin{aligned} \text{Prob} \quad & \{\exists R \in \mathcal{F} \text{ s.t. } R \text{ is not covered by a random sequence at time } mt\} \\ \leq \quad & \sum_{R \in \mathcal{F}} \text{Prob}\{ R \text{ is not covered by a random sequence at time } mt \} \\ \leq \quad & 2^{-m} |\mathcal{F}| \quad \leq \quad 2^{n^2 - m}. \end{aligned}$$

Hence, if we choose $m > n^2$, then the probability for existence of a sequence that does not cover all polygons in $\mathcal{F}$ within an $mt$ period is less than one; i.e., there exists such a sequence that does guarantee covering of all polygons in $\mathcal{F}$, and hence there is a $(2n^4 \log n)$-time sequence of angles that is a UTSA for $\mathcal{F}$. □

Note that finding a universal sequence of length $O(4^n)$ is easy: just traverse the ternary tree of height $n$ with the starting point as the root and with four neighbors to each vertex, each representing a turning angle from $\{0, \pi/2, \pi, 3\pi/2\}$. Backtracking is possible thanks to the "compass" that our robot has. Clearly, not all steps will be of length $r$, because of walls and obstacles, but eventually all squares will be reached.

### *4.4. An Example*

As an example, consider a square room of 2-m sides and a robot of 40-cm radius. We have $a/r = 5$ and $2.5 \leq \rho \leq 5$, so $m = 24$ and $n = 176$. Substituting in Theorem 3, one gets $\mathbf{E}[T] \leq 10{,}976$. Assuming a robot's speed of 10 cm/sec, 10,976 steps take about 732 min. With the MAC algorithm and the same setting, our bound is 90 steps or 6 min. In reality, a single MAC step is much slower than a single PC step, since the MAC robot has to spend enough time along the path to properly mark and sense it. It should also be noted that our upper bound on $\mathbf{E}[T^P C]$ is probably loose, since simulations show (for the above case) an average cover time of 234 steps with standard deviation of 74 steps, which, in the above setting, amounts to an average cover time of 15.5 min, i.e., about 2.6 times the MAC period. A plot of the average cover time and standard deviation against the number of robots is depicted in Figure 16.

## 5. MAC-PC: A Hybrid Algorithm

The MAC algorithm has the advantages of a guaranteed coverage within time $O(A/a)$ and awareness of completion, but is vulnerable to physical problems, e.g., vanishing of the traces with time and sensory errors. The random PC procedure, on the other hand, is almost free of dependency on sensors (an exception is its need to identify a collision), but has no awareness of completion and is slower: its average cover time is at most $O(Ad/a)$ for region of diameter $d$. In this section, we suggest a combination of the two methods that gives a reasonable trade-off between performance and robustness.

One way to combine the algorithms is to have the robot follow a random PC procedure; i.e., a new direction is selected at random in each step, but the radius of coverage (and step size) is $r' > r$, using the (short-lived) marks to cover a circle of such an extended radius. Formally, the rule of motion is similar to our previous PC algorithm:

/* MAC-PC - Semi-Probabilistic Covering */
/* $r$ = covering radius; */
/* $r'$ = extended coverage radius, */
/* depending on the trace's lifetime */

10. Note that in our setting, the time taken by a sequence of steps $z_1, z_2, \ldots, z_k$ is not necessarily equal to $k$, the number of steps; rather, it is equal to $\sum_{i=1}^{k} \mu(z_i)^2$.

11. A graph is $k$-regular if exactly $k$ edges emanate from every vertex. It is *edge-labeled* if the edges emanating from each vertex are numbered in some order.

12. This is due to Markov's inequality (Motwani and Raghavan 1995, p. 46), which says that if $Y$ is a nonnegative random variable, then for all $\epsilon \in \mathbf{R}^+$ it holds that $\text{Prob}\{Y \geq \epsilon\} \leq \frac{\mathbf{E}[Y]}{\epsilon}$. Substituting $\epsilon = 2\mathbf{E}[Y]$, one finds that after twice the expected cover time, the probability of full coverage is at least $1/2$.

Rule MAC-PC($z$: current location)
A) cover $B_{r'}(z)$ by applying a local MAC process from $z$;
B) choose a random neighbor $w$ from $R \cap C_{r'}(z)$;
C) go to $w$;
end MAC-PC.

See Figure 11 for an example.

Note that each step of the MAC-PC algorithm takes $O((r'/r)^2)$ units of time. Hence, using Theorem 3 (but this time with squares of side $r'/\sqrt{2}$ rather than $r/\sqrt{2}$), the expected time to cover a region by the hybrid algorithm is bounded from above:

$$\mathbf{E}\left[T^{\text{MAC-PC}}\right] = O\left(\frac{\rho(r/r')A}{r'^2}(r'/r)^2 \log\left(\frac{A}{r'^2}\right)\right)$$
$$= O\left(\frac{A\rho}{r'r}\log\left(\frac{A}{r'^2}\right)\right),$$

which is more than $O(r'/r)$ times faster than the bound on a simple PC.

## 6. Simulation and Experiments

A simulation program (Wagner 1997) that animates the algorithms in this paper was written in JAVA[13] (see the examples in Figs. 12 and 13). In addition to visualization of the dynamics of the algorithms, our simulator can also be used to

13. The reader is encouraged to try the simulator, which is web accessible via http://www.cs.technion.ac.il/~ wagner/pub/mac.html.

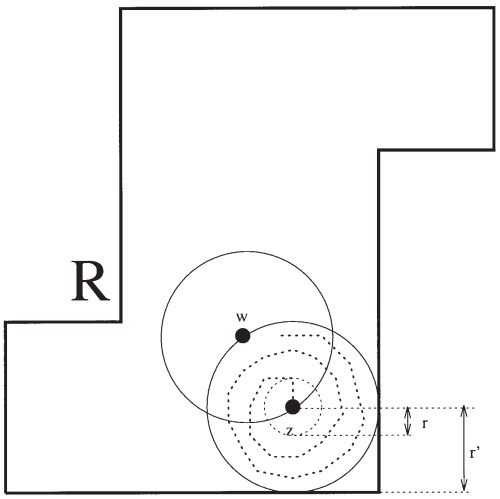


Fig. 11. A hybrid of MAC and PC: the covering radius is widened by using short-lived trails to traverse the $r'$ circle around $z$, as compared to the $r$ circle covered by the conventional PC rule. After covering $B_{r'}(z)$, the robot goes to $w$, a randomly chosen point at a distance $r'$ from $z$. Working this way, the average performance improves significantly, while using only short-lived traces.

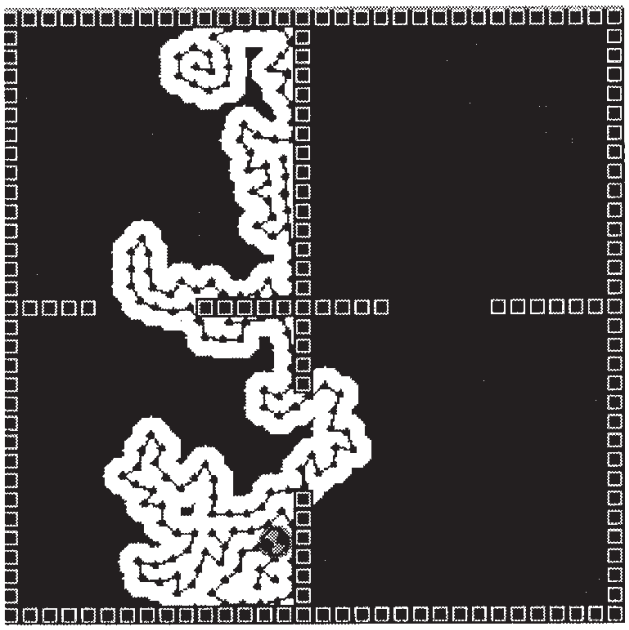

Fig. 12. A simulation of the MAC algorithm with one robot. The marks appear as a thin line, while gray denotes the uncovered ("dirty") region.

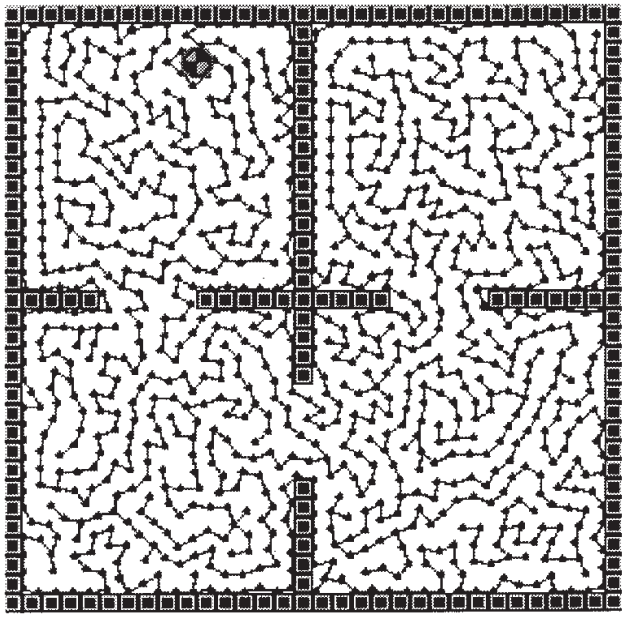

Fig. 13. Upon completion, the final trace induced by MAC is a tree that spans the region.

study those properties of the algorithms for which we have no satisfactory analysis; e.g., the dependency of cover time on the number of robots. In Figure 16, a comparison of the deterministic MAC and the random PC algorithms is depicted, by way of the times of complete coverage of a $30 \times 30$ room, with and without walls in it (there is a one-unit opening in the wall; see Figs. 14 and 15), and plotted against the number of robots. Note that in all MAC simulations we assume that the trails remain detectable during the whole time of execution; in the worst case (e.g., that of a robot starting near the edge of a very long and narrow corridor), this assumption is required to guarantee covering. For the PC algorithm, the cover times were averaged over 20 simulation runs, and the standard deviation was plotted as well. The simulations show that MAC is much faster, and that both algorithms benefit from increasing the population of robots, up to a point where robots begin to disturb rather than help each other.

## 7. Summary

We have analyzed two ways of covering a continuous region, using the deterministic MAC and the random PC. The setting of continuous space is more relevant to robotics than the discrete structure of graphs, since robots move continuously, and even if a discrete partition is dictated by some external signs (e.g., a tiled floor), it is still hard for a low-cost robot to precisely identify those signs. The problem of continuous covering has various implications for both theory and practice. The analysis suggested in this paper can serve as an inspiration for further research in several directions, some of which are described below:

1. **Covering without positioning.** In our MAC algorithm, we assume that the robot is able to either follow a previously laid trace, or recall a previous location from its memory, based on some positioning mechanism (e.g., odometric positioning or GPS). These positioning systems are both expensive and error-prone. It seems, however, that to guarantee coverage, we only need our positioning system to be unique, and not necessarily precise; i.e., if the position is $z$ and our system returns $P(z)$ as a position, we only need to have $P(z) \neq P(z')$ if $z \neq z'$. The open question here is how to define the precise conditions on the positioning function $P(z)$ that allow a MAC-like algorithm to cover the region.
2. **Better mode of cooperation.** In our multi-robot setting, we just add robots and let them all follow the same rule. It is intriguing to see what will happen if a more significant communication is enabled; e.g., considering the MAC algorithm, if a robot that has finished its mission could get a call from a robot that is still working and come to help. Such an approach may yield a better speedup by increasing the number of robots.
3. **Cooperating PC robots.** As we just add robots and let them all follow the same PC rule, it would be intriguing to see what would happen if a more significant communication were enabled; e.g., if a collision with another robot or with the wall would make the future steps biased against the (alleged) location of other robots/walls.

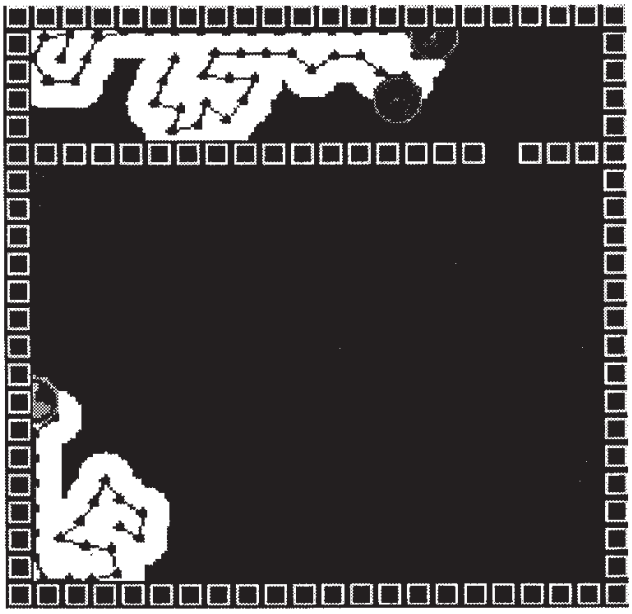

Fig. 14. A case of one wall for MAC/PC simulations.

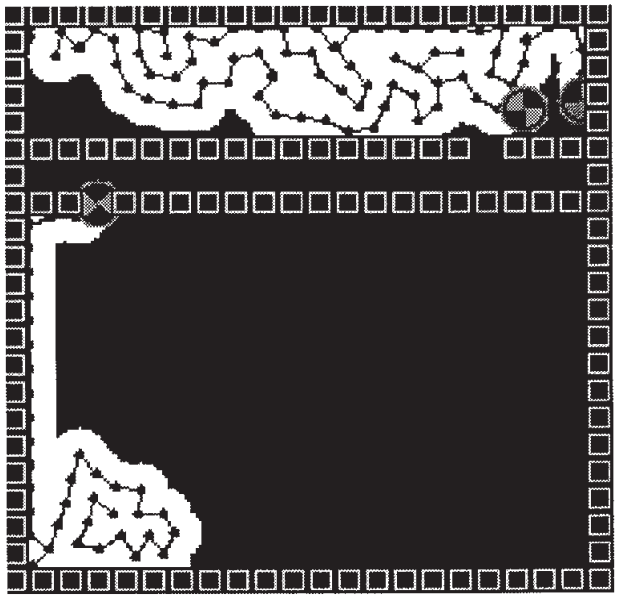

Fig. 15. A case of two walls for MAC/PC simulations.

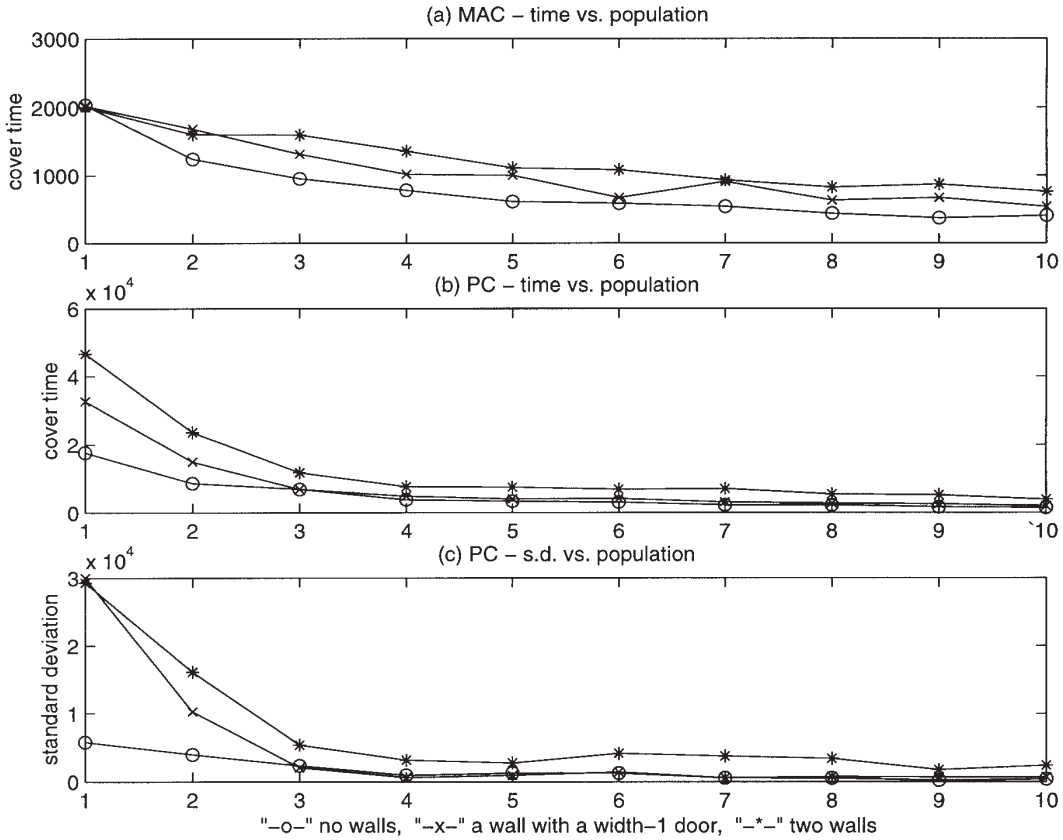


Fig. 16. A comparison of the deterministic MAC and the random PC algorithms. Times of complete coverage of a $30 \times 30$ room, with one wall ("-x-" points), two walls ("-*-" points), and without walls ("-o-" points), by MAC- (a) and PC-simulated (b) robots are plotted against the number of robots. Note the difference in scale of the $y$ axes. For the PC algorithm, the cover times were averaged over 20 simulation runs, and the standard deviation is plotted (c). The simulator is accessible via the web at http://www.cs.technion.ac.il/~ wagner/pub/mac.html.

4. **Finding a "short" universal traversal sequence of angles.** We have shown the existence of a polynomial-length universal sequence of angles (UTSA) for gridded polygons. However, we do not know how to find one. The similar question for graphs is also wide open, with the only exceptions (known to us) being paths and cycles (Bridgland 1987; Bar-Noy et al. 1989). Intuitively, one may think that finding a UTSA in our case is easier, since the robot is assumed to have a kind of "compass," while in the UTS problem for graphs, edges are arbitrarily ordered.

5. **The minimum memory needed to deterministically cover a region.** Counting the total amount of memory used as marks on the floor, our MAC robots need $O(n)$ memory ($n$ being equal to $A/a$). Can an algorithm be devised that covers a finite region with a constant amount of memory?

6. **Parametric step-coverage problem.** If a graph is embedded such that each edge is a zero-width line with an integer length, then a minimum covering with radius $r \to 0$ is equivalent to a Euler path[14] in the graph, if one exists. Hence both Hamilton and Euler paths are special cases of the covering path. Thus we have shown that the question, "Is there a 0.5-covering of length $|\phi(G)|$ in a grid-graph $G$," is hard, while the other extreme, "Is there a 0-covering of length $|E(G)|$ in a grid-graph $G$," is easy; so how hard is the question "Is there an $r$-covering of length $L$, $|\phi(G)| \leq L \leq |E(G)|$ in a grid-graph $G$," where $0 < r < 0.5$?

## Appendix: Potential Difference across a Uniformly Resistive Circle

### *Proof of Proposition 1*

Consider a circle of radius $\mu$ and unit sheet resistance, and assume that a current of $I_0$ A per unit area is uniformly injected

14. A *Euler path* is a tour of the graph in which each edge is traversed exactly once. A graph is *Eulerian* if such a path exists.

into the circle. We seek for the average potential difference (or "voltage drop") between the center of the circle and its circumference, defined by

$$\overline{\phi(0) - \phi(\mu)} = \frac{1}{2\pi} \int_{\theta=0}^{2\pi} (\phi(0) - \phi(\mu e^{i\theta}))d\theta.$$

Consider a ring of radius $u$ and infinitesimal width $du$ (see Fig. 17). We know (from the theorem of divergence) that since there are no sources or sinks of current on the surface, all the current injected into the $u$ circle should flow out across its boundary and into the ring. This amount of current is $I_0 \pi u^2$. Let us denote by $I(u,\theta)$ the centrifugal current flowing at $ue^{i\theta}$ in direction $\theta$, by $d\phi(u,\theta)$ the voltage drop between the inner and outer edges of an infinitesimal trapezoid of the ring, and by $\overline{d\phi(u)}$ the average voltage drop across the ring. One can now write

$$\begin{aligned} \overline{d\phi(u)} &= \frac{1}{2\pi} \int_{\theta=0}^{2\pi} d\phi(u,\theta)d\theta \\ &= \frac{1}{2\pi} \int_{\theta=0}^{2\pi} \frac{I(u,\theta)dud\theta}{ud\theta} \end{aligned}$$

(the resistance of a rectangle is length/width)

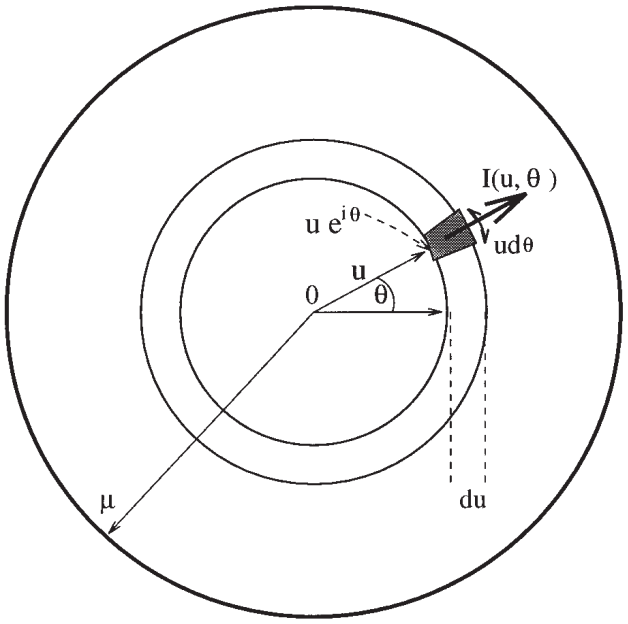


Fig. 17. An infinitesimal ring within a circle. The average voltage drop across the ring is obtained by integrating over small trapezoids like the one in gray, through which the centrifugal current $I(u,\theta)$ is flowing.

$$\begin{aligned} &= \frac{du}{2\pi u} \int_{\theta=0}^{2\pi} I(u,\theta)d\theta \\ &= \frac{du}{2\pi u} \pi u^2 I_0 = \frac{I_0 u du}{2} \quad . \end{aligned}$$

Note that the voltage drop across the ring due to the current flowing into the ring itself is proportional to the product of this current, $(o((u+du)^2 - u^2) = o(udu))$, and the ring's resistance, $((o(du/u))$, and hence it is $o((du)^2)$, and vanishes in integration. Thus, the total voltage difference can be found by integrating along $u$:

$$\overline{\phi(0) - \phi(\mu)} = \int_{u=0}^{\mu} \overline{d\phi(u)}du = \int_{u=0}^{\mu} \frac{I_0 u}{2} du = \frac{I_0 \mu^2}{4} \quad .$$

□

## Acknowledgment

We would like to thank the anonymous referees for their comments and David Aldous for his advice with respect to the results in his work (Aldous 1991).

## References

Aldous, D. J. 1991. Threshold limits for cover times. *J. Theoret. Probabil.* 4(1):197–211.

Aldous, D. J., and Fill, J. A. 1996. *Reversible Markov chains and random walks on graphs.* A book in preparation, available via the Web at http://www.stat.Berkeley.EDU/users/aldous/book.html.

Aleliunas, R., et al. 1979 (Oct. 29–31, San Juan, Puerto Rico). Random walks, universal traversal sequences, and the complexity of maze problems. *Proc. of the 20th Annual Symp. on Found. of Comp. Science*, pp. 218–223.

Arkin, E. M., and Hassin, R. 1994. Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Math.* 55:197–218.

Baeza-Yates, R., Culberson, J. C., and Rawlins, G. J. E. 1993. Searching in the plane. *Inform. Computation* 106(2):234–252.

Baeza-Yates, R., and Schott, R. 1995. Parallel searching in the plane. *Computational Geometry Theory Applications* 5:143–154.

Balch, T., and Arkin, R. C. 1994. Communication in reactive multiagent robotic systems. *Autonomous Robots* 1(1):27–52.

Barnes, G., and Feige, U. 1993. (May 16-18, San Diego, CA). Short random walks on graphs. *Proc. of the 25th ACM Symp. on Theory of Computing*. New York: ACM, pp. 728–737.

Bar-Noy, A., et al. 1989. Bounds on universal sequences. *SIAM J. Comp.* 18(2):268–277.

Berger, M. 1987. *Geometry II*. Berlin-Heidelberg: Springer-Verlag.

Bessiere, P., Ahuactzin, J. M., Talbi, E. G., and Mazer, E. 1995. The "Ariadne's clew" algorithm: Global planning with local methods. *Algorithmic Foundation of Robotics.* Wellesley, MA: A. K. Peters, pp. 39–47.

Bollobas, B. 1990. *Graph Theory: An Introductory Course.* Berlin-Heidlberg: Springer-Verlag.

Borenstein, J., Everett, H. R., and Feng, L. 1996. *Navigating Mobile Robots: Systems and Techniques.* Wellesley, MA: A. K. Peters.

Borenstein, J., and Feng, L. 1996. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Trans. Robot. Automat.* 12(6):869–880.

Bridgland, M. F. 1987. Universal traversal sequences for paths and cycles. *J. Algorithms* 8(3):395–404.

Broder, A. Z., Karlin, A. R., Raghavan, P., and Upfal, E. 1994. Trading space for time in undirected $s - t$ connectivity. *SIAM J. Comp.* 23(2):324–334.

Chandra, A. K. et al. 1997. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity* 6:312–340.

Chin, W. P., and Ntafos S. 1986 (June 2–4, Yorktown Heights, NY). Optimum watchman routes. *Proc. of the 2nd Annual Symp. on Computational Geometry*, pp. 24–33.

Deng, X., and Mirzaian, A. 1996. Competitive robot mapping with homogeneous markers. *IEEE Trans. Robot. Automat.* 12(4):532–542.

Do-Carmo, M. P. 1976. *Differential Geometry of Curves and Surfaces.* Englewood Cliffs, NJ: Prentice-Hall.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. 1991. Robotic exploration as graph construction. *IEEE Trans. Robot. Automat.* 7(6):859–864.

Enge, P. et al. 1996. Wide-area augmentation of the global positioning system. *Proc. IEEE* 84(8):1063–1088.

Erdmann, M. 1992. Randomization in robot tasks. *Intl. J. Robot. Res.* 11(5):399–436.

Even, S. 1979. *Graph Algorithms.* Rockville, MD: Computer Science Press.

Feller, W. 1968. *An Introduction to Probability Theory and Its Applications*, third ed., vol. 1. New York: Wiley.

Gage, D. W. 1993 (Sept. 9–10, Boston, MA). Randomized search strategies with imperfect sensors. *Proc. of SPIE Mobile Robots VIII*, vol. 2058, pp. 270–279.

Giralt, G., Weisbin, C. (eds.) 1995. *Autonomous Robots* 2:259–362 (special issue on autonomous robots for planetary exploration).

Graham, R. L., Knuth, D. E., and Patashnik, O. 1991. *Concrete Mathematics*. Reading, MA: Addison-Wesley.

Hall, P. 1988. *Introduction to the Theory of Coverage Processes.* New York: Wiley.

Held, M. 1991. *On the Computational Geometry of Pocket Machining*, a vol. of *Lecture Notes in Computer Science*. Berlin-Heidelberg: Springer-Verlag.

Hert, S., Tiwari, S., and Lumelsky, V. 1996. A terrain-covering algorithm for an AUV. *Autonomous Robots* 3(2/3):91–119.

Hofner, C., and Schmidt, G. 1995 Path planning and guidance techniques for an autonomous mobile cleaning robot. *Robot. Autonomous Sys.* 14:199–212.

Itai, A., Papadimitriou, C. H., and Szwarefiter, J. L. 1982. Hamilton paths in grid graphs. *SIAM J. Comp.* 11(4):676–686.

Kaplan, W. 1984. *Advanced Calculus*, 3rd ed. Reading, MA: Addison-Wesley.

Kershner, R. 1939. The number of circles covering a set. *Am. J. Math.* 61:665–671.

Kuipers, B., and Byun, Y. T. 1981. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robot. Autonomous Sys.* 8:47–63.

LaValle, S. M., and Hutchinson, S. A. 1996. Evaluating motion strategies under nondeterministic or probabilistic uncertainties in sensing and control. *Proc. of the 1996 IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 3034–3039.

Lovasz, L. 1996. Random walks on graphs: A survey. In Miklos, D., Sos, V. T., and Szony, T. (eds.): *Combinatorics, Paul Erdös is Eighty*, vol. 2. Budapest: Janos Bolyai Mathematical Society, pp. 353–398.

Lumelsky, V. J., and Stepanov, A. A. 1987. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica* 2:403–430.

Matthews, P. 1989. Covering problems for Brownian motion on spheres. *Annals Probability* 16(1):189–199.

Motwani, R., and Raghavan, P. 1995. *Randomized Algorithms.* Cambridge: Cambridge University Press.

Ntafos, S. 1992. Watchman routes under limited visibility. *Computational Geometry Theory Applications* 1:149–170.

Parker, L. E. 1996. On the design of behavior-based multi-robot teams. *Adv. Robot.* 10(6):547–578.

Russell, R. A. 1995. Laying and sensing odor markings as a strategy for assisting mobile robot's navigation tasks. *IEEE Robot. Automat. Mag.* 2(3):3–9.

Russell, R. A. 1997. Heat trails as short-lived navigational markers for mobile robots. *Proc. of the 1997 IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 3534–3539.

Sagan, H. 1994. *Space-Filling Curves.* New York: Springer-Verlag.

Singh, S., Hines, E. L., and Gardner, J. W. 1996. Fuzzy neural computing of coffee and tainted-water data from an electronic nose. *Sensors Actuators (B)* 30:185–190.

Traub, J. F., Wasilkowski, G. W., and Wozniakowski, H. 1988. *Information-Based Complexity.* San Diego, CA: Academic Press.

Virgil, 1971. *The Aeneid of Virgil,* trans. by A. Mandelbaum. New York: Bantam Books.

Wagner, I. A. 1997. MAC, PC, and other gadgets—a JAVA online simulator. Available via the World Wide Web at http://www.cs.technion.ac.il/~ wagner/pub/mac.html.

Wagner, I. A., and Bruckstein, A. M. 1997. Cooperative cleaners: A study in ant robotics. In Paulraj, A., Roychowdhury, V., and Schaper, C. D. (eds.): *Communications, Computation, Control, and Signal Processing: A Tribute to Thomas Kailath.* The Netherlands: Kluwer Academic.

Wagner, I. A., Lindenbaum, M., and Bruckstein, A. M. 1996 (June 10–12, Jerusalem). Smell as a computational resource—a lesson we can learn from the ant. *Proc. of the 4th Israeli Symp. on the Theory of Comp. and Sys.* (Also Technical Report CIS-9610, Center for Intelligent Systems, Technion, Haifa, April 1996.)

Wagner, I. A., Lindenbaum, M., and Bruckstein, A. M. 1997 (July 28, Providence, RI). On-line graph searching by a smell-oriented vertex process. *Working Notes from the AAAI'97 Workshop on On-Line Search*, pp. 122–125.

Yaguchi, H. 1996. Robot introduction to cleaning work in the East Japan Railway Company. *Adv. Robot.* 10(4):403–414.