Path-Planning for Robust Area Coverage: Evaluation of Five Coordination Strategies *

Manuel Mazo Jr. **[1] and Karl Henrik Johansson[2]

[1] Department of Electrical Engineering, University of California,Los Angeles, California, USA
mmazo@ee.ucla.edu
[2] School of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden
kallej@ee.kth.se

Abstract. Robust area coverage is the problem of driving the footprint of a mobile robot over all the points of a given region in an efficient manner even when noise is present in actuators and sensors. This problem is a common challenge in many applications, including automatic lawn mowing and vacuum cleaning. In this paper a robot with uncertain heading is studied. Five control strategies based on event-triggered position measurements available when the vehicle intersects the boundary of the area to be covered are compared. It is shown that the performance depends heavily on the maximum size of the heading error. The results are evaluated through extensive Monte Carlo simulations for the coverage of a square-shaped cell. An experimental implementation is also presented.

1 Introduction

Coverage robotics is on moving a mobile platform in order to cover a surface by robot's bottom or sensor range.

Applications of coverage robotics range from mine, search-and-rescue, surveillance and map generation to painting machines and snow removing vehicles. Recent commercial implementations in consumer products include automatic vacuum cleaners (Electrolux, 2001) and automatic lawn mowers (AB, Husqvarna, 1998).

In this paper we focus on the coverage problem of efficiently coordinate a mobile robot's footprint to cover a specified area. Coverage through sensing, such as the art-gallery problem (Gonzalez-Banos & Latombe, 2001) and various sensor network problems (Poduri & Sukhatme, 2004) are not treated.

* The work by M. Mazo Jr. was partly supported by the Spanish Ministry of Science and Education. The work by K.H. Johansson was partly supported by the Swedish Foundation for Strategic Research through an Individual Grant for the Advancement of Research Leaders and by the Swedish Research Council.

** Corresponding author

1.1 Contribution

The main contribution of this paper is to introduce and study an area coverage problem for a mobile robot with an uncertain vehicle model. Based on this model and the assumption that position measurements are only available at the boundary of the area to be covered, five control strategies are analyzed through extensive simulations and experiments. This setup is quite realistic, for example, for mobile robots that suffer from unreliable position readings and limited sensor capacity. By appropiate marking of the boundary of the area to be covered, it is possible to navigate without localization information. The efficiency of a strategy is measured by the total number of turns needed to cover the area, cf., (Huang *et al.*, 1986). This is based on the natural assumptions that for mobile robots and other vehicles, turns are costly due to the need to decelerate, turn, and accelerate. The proposed closed-loop path-planning system for the area coverage is modeled as a hybrid automaton. In this way, it is possible to have a low-order model, which still can capture the essentials of the coordination problem. The efficiency to use hybrid control in robotics is also illustrated in time-optimal tracking control problems for Dubin's vehicle (Balluchi & Souères, 1996).

The proposed path-planning strategies are evaluated by comparing the number of turns for various degrees of heading uncertainty. It is shown that for large uncertainties, a randomized strategy is the best one, which is intuitive since the system state does not reveal much information in that case. For small uncertainties, a modified boustrophedon-path strategy sweeping the area by a simple back-and-forth motion is sufficient. The interesting case, however, is for uncertainties of middle range. We present three robust path-planning algorithms that outperform the randomized and the Boustrophedon strategies for mobile robots with middle range heading uncertainty.

The tools used for computing the commands are based on recent computational geometry software (Vatti, 1992), (Murta, 2003). The complexity of the coverage algorithms is not studied in this paper. In general, one can probably say, however, that the presented solutions do not scale well. A complexity analysis of the algorithms used in the geometrical operations can be found in (Greiner & Hormann, 1998). In this context, we also remind of the art gallery problem, which has been extensively studied also regarding algorithmic complexity (Gonzalez-Banos & Latombe, 2001).

1.2 Related Work

In the past years the coverage problem has received much attention and several solutions have been proposed in the literature including single and multi-robot solutions, see (Choset, 2001) for a comprehensive survey.

An early reference to the coverage problem can be found in the work of Oommen et al. (Oommen *et al.*, 1987), which presents an algorithm for covering an unknown planar region with convex polygonal obstacles. This work is extended to the case of non-convex polygonal obstacles by Rao and Iyengar (Rao & Iyengar, 1990). Hert and Lumelsky (Hert *et al.*, 1996) treat the problem of non-polygonal surfaces and generalize to a three-dimensional environment. Similar approaches are presented by Choset (Choset, 2000), (Choset & Pignon, 1997) who proposes a decomposition of the region to be covered into smaller pieces. These pieces are then easily covered by a local coverage strategy, and the collection of pieces are connected by solving a traveling salesman problem with the pieces as nodes of the adjacency graph. These techniques are known as exact cellular decomposition. The trapezoidal cellular decomposition (Latombe, 1991) (also known as the slab method (Preparata & Shamos, 1985)) is an earlier technique of this kind, which is outperformed by Choset's solution. Hert and Lumelsky's algorithm can be also included in the cellular decomposition frame. The latest advances alon this line of research include the Minimal Sum of Altitudes algorithm by Huang (Huang, 2001), proposing better paths than the boustrophedon decomposition by taking into account the sweep direction and using the number of turns as a measurement of efficiency; and Choset et al. (Choset *et al.*, 2000) expanding his decomposition technique to higher dimensional spaces. Other algorithms have dealt with this problem in discretized environments producing complete algorithms both for single robot (Zelinsky *et al.*, 1993) and for multiple robots (Kurabayashi *et al.*, 1996). Behavior-based algorithms for multi-robots coverage are studied in (Balch & Arkin, 1995), (MacKenzie & Balch, 1996), (Svennebring & Koenig, 2004). Actuator and sensor errors have not been considered much in the area coverage literature, though they seem to play an important role on the performance in many applications. One of the few contributions not assuming perfect localization information is a study of ant-like robots by Svennebring and Koening (Svennebring & Koenig, 2004). Our work focuses on a simple model to capture the main uncertainty in the problem. Contributions focused on floor vacuum cleaning, harvesting of large fields and the generation of underwater mosaicked images include (Colegrave & Branch, 1994), (Ollis & Stenz, 1996), (Hert *et al.*, 1996).

1.3 Outline

The outline of the paper is as follows. The area coverage problem is formulated in Section 2. Five path-planning strategies for solving the problem is presented in Section 3. We present a brief study of the implementation complexity in Section 4. In Section 5 it is shown by extensive Monte Carlo simulations that the preferable path-planning strategy depends on the error bound of the steering actuator. Experimental results are presented in Section 6, where an implementation on a mobile robot is demonstrated. Section 7 concludes the paper.

2 Problem Formulation

Consider the problem of covering the set

$$\Omega = [0,L] \times [0,L] \subset \mathbb{R}^2, \quad L \gg 1,$$

by a square vehicle, as illustrated in Figure 1. The vehicle covers a unit square, which is positioned with its upper-right corner at coordinate (x,y). For simplicity, we assume that the vehicle starts in the lower-left corner $(x(0), y(0)) = (1,1)$. At time $t \geq 0$, the vehicle covers the set

$$c(t) = [x(t)-1, x(t)] \times [y(t)-1, y(t)].$$

The accumulated covered set is denoted

$$C(t) = \bigcup_{s\in[0,t]} c(s).$$

The vehicle is a unicycle specified as

$$\begin{aligned}
\dot{x}(t) &= \cos\theta(t) \\
\dot{y}(t) &= \sin\theta(t) \\
\dot{\theta}(t) &= \omega(t) \\
\dot{v}(t) &= F(t)/m \\
\dot{\omega}(t) &= \tau(t)/J,
\end{aligned} \tag{1}$$

where the force F and torque τ are the control inputs. Suppose each wheel is independently actuated with force F_1 and F_2, respectively. The force and torque can then be expressed as

$$\begin{pmatrix} F \\ \tau \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{2}{\ell} & \frac{2}{\ell} \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}, \tag{2}$$

where ℓ is the distance between the wheels.

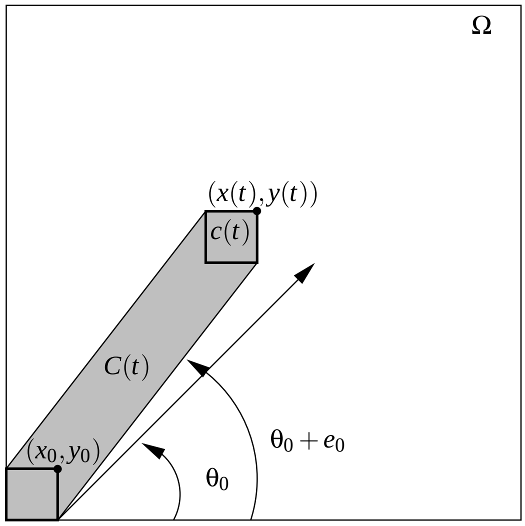

Fig. 1. Area coverage problem. A square vehicle with uncertain dynamics should cover the area of Ω as fast as possible.

Consider two possible maneuvers: the robot is moving with unit velocity $v = 1$ straight ahead in a constant heading direction θ; or the robot is standing still ($v = 0$) but turning around (changing its heading direction). From Equations 1 and 2, we note that the first maneuver can be made by setting $F_1 = F_2$ and using a linear controller to enforce the velocity to be about one. For the second maneuver, we use $F_1 = -F_2$ and another linear controller to drive the heading angle to the desired turning angle. Based on these two maneuvers, we formulate the simplified model for the robot movement

$$\begin{aligned} \dot{x}(t) &= \cos(\theta(t) + e(t)) \\ \dot{y}(t) &= \sin(\theta(t) + e(t)), \end{aligned} \tag{3}$$

where $\theta \in [-\pi, \pi)$ is the controlled heading and e an unknown angular error. The error, which thus affects the actuation of the control, is supposed to be bounded by a known constant $\varepsilon \in [0, \pi)$, and can be seen as an static error inherited from the turning maneuver.

The vehicle localization is constrained, such that the vehicle position is known only at moments when the vehicle hits the boundary of Ω, i.e., $t > 0$ such that $c(t) \cap \partial\Omega \neq \emptyset$. This can be implemented in practice by marking the boundary in a suitable way; compare current systems used for automatic cleaning robots (Electrolux, 2001) and lawn

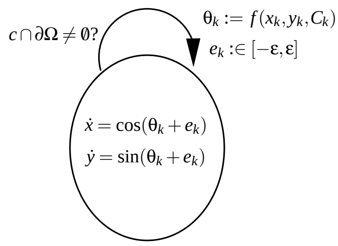

Fig. 2. A hybrid automaton describing the area coverage path-planning problem. When the guard condition $c \cap \partial\Omega \neq \emptyset$ is enabled (i.e., the vehicle coverage intersects the boundary of Ω), a discrete event takes place. At the event, the control θ_k and error e_k are updated according to a path-planning law f and an error set $[-\varepsilon, \varepsilon]$, respectively.

movers (AB, Husqvarna, 1998). The path-planning strategies studied in the paper are limited to piecewise constant controls triggered by the events $c(t) \cap \partial\Omega \neq \emptyset$, which corresponds to moments when the vehicle turns. The error e is piecewise constant and should be interpreted as the uncertainty in the actuation of the turning angle. We suppose that the turning events are separated in time and denoted $0 = t_0 < t_1 < \dots$. The control θ at turn k is denoted θ_k and the corresponding error e_k. We suppose that $\theta_k + e_k$ never drives the vehicle outside Ω, i.e., for all $t \geq 0$ we have $c(t) \subset \Omega$.

In order to efficiently cover the area of Ω, denoted $A(\Omega) = \int_\Omega dz$, it is reasonable to try to minimize the total number of turns $N > 1$ made by the vehicle to complete the coverage, cf., (Huang *et al.*, 1986), (Choset, 2001). The feedback controls θ_k, $k = 0, 1, \dots, N$, can be written as

$$\theta_k = f(x_k, y_k, C_k),$$

where $(x_k, y_k) = (x(t_k), y(t_k))$ is the position at the turning point and $C_k = C(t_k)$ is the total covered set up till time t_k.

The closed-loop system can be described as the hybrid automaton in Figure 2, which visualizes the dynamics in an intrinsic way. When the guard condition $c(t) \cap \partial\Omega \neq \emptyset$ is fulfilled, a discrete-event is generated. It updates the control θ and the error e according to the indicated reset maps. A path-planning law f that solves the coverage problem in N turns corresponds to a family of hybrid trajectories, which each consists of N straight lines.

The considered hybrid differential game problem is now to find a feedback path-planning law f that minimizes N, given hard constraint on the error $|e_k| < \varepsilon$. The authors are not aware of a general solution to this path-planning problem. Some greedy and approximate solutions are presented in the next section.

3 Path-Planning Strategies for Area Coverage

Five feedback path-planning strategies for area coverage are presented in this section. They are denoted *Receding Horizon (RH)*, *Robust Receding Horizon (RRH)*, *Opportunistic Receding Horizon (ORH)*, *boustrophedon*, and *randomized*. The first three of them are "greedy" in the sense that they try to maximize the area covered between two turns given different constraints. The fourth strategy is heuristic and basically mimics a traditional way of covering an area when there are no actuator errors; a boustrophedon path (Choset, 2001), (Choset & Pignon, 1997). The fifth control strategy is a randomized solution, which is inspired by commercial implementations in automatic vacuum cleaners and lawn movers (Electrolux, 2001), (AB, Husqvarna, 1998).

Before discussing the five strategies, let us elaborate on an optimal solution to the proposed problem. A global solution to minimizing the number of turns given an initial condition is a complex computational problem in general because of the unknown angular error. The problem of minimizing the number of turns to cover a portion of the goal region is essentially the same as maximizing the area covered after a certain number of unknown turns. As we have an unknown angular error in our moving direction the optimization should weight the covered areas with the probability of that certain direction. If we assume the distribution of this error $p(e)$ to be known and bounded, the optimization problem can be stated as:

$$\max_{\theta_1,\ldots,\theta_k} \int_{-\varepsilon}^{\varepsilon} \cdots \int_{-\varepsilon}^{\varepsilon} A(C_k) \prod_{i=1}^{k} p(e_i) de_i$$

Trying to solve the problem directly presents a major problem: our optimization objective does not have a closed algebraic form, thus its solution must be achieved by exploration of the different input combinations. To do this exploration we can discretize both the possible command directions θ, and the p.d.f. of e, so that we also have a finite number of possible error values. It is obvious that with increasing N, the complexity of the problem explodes, and much faster when we use more discrete states to describe θ and e. To cope with the complexity problem with increasing N, we can apply a *receding horizon* optimization policy as:

$$\max_{\theta_k} \sum_{k=1}^{\hat{N}} \sum_{e_k \in [-\varepsilon,\varepsilon]} A\left(B(x_k, y_k, \theta_k + e_k \cup C_k)\right) p(e_k)$$

where

$$B(x,y,\theta) = \bigcup_{z \in \ell(x,y,\theta) : \bar{c}(z) \subset \Omega} \bar{c}(z)$$

This kind of policy can be justified as a trade-off between optimality and complexity. Also a justification for using such approach is the unknown value of the N number of turns to achieve a certain target percentage of the total coverage, or the total coverage itself. Formally the optimal solution would then be given by solving the receding horizon optimization:

$$\lim_{\hat{N}\to\infty} \max_{\theta_k} \sum_{k=1}^{\hat{N}} \sum_{e_k\in[-\varepsilon,\varepsilon]} A\left(B(x_k, y_k, \theta_k + e_k \cup C_k)\right) p(e_k)$$

Yet seems reasonable to think that with a $p(e)$ with high bounds the loss of optimality when reducing the number of steps ahead in the receding horizon is much smaller than in the ideal case in absence of error. Even when the error has tight bounds, if the number of turns necessary to achieve a certain percentage of global coverage is big enough, optimizing over $\hat{N}$ steps instead of the unknown total N or a preset higher bound $\overline{N}$ will result in a small loss of optimality. Hence in the quest for reducing complexity, knowing that in a general case with a high $\overline{N}$ we will not be too far from the optimal solution, we can set up $\hat{N} = 1$, it is, use a one-step-ahead optimization. Obtaining at each step an optimization problem like:

$$\max_{\theta_k} \sum_{e_k\in[-\varepsilon,\varepsilon]} A\left(B(x_k, y_k, \theta_k + e_k \cup C_k)\right) p(e_k)$$

As for the problem of capturing the influence of the error, as we will present in the description of the three first strategies developed, we can use different approximations of the above posed sum weight with the probability.

This way we have achieved to reduce greatly the complexity of the optimization by exploration of the algorithm at the cost of obtaining only sub-optimal solutions.

Figure 3 shows a comparison of how the first three path-planning strategies are derived. The snapshot is taken at turn $k = 2$. The current coverage $c(t_2)$ of the vehicle is marked by a small square. The gray area corresponds to the accumulated coverage $C(t_2)$. At this moment, the algorithms maximize the area to be covered till turn $k = 3$, i.e., search for the best control θ_2 over the interval $[-\pi, \pi)$. Figure 3 shows areas for $\theta_2 = -\pi/4$. How these are derived is further described below.

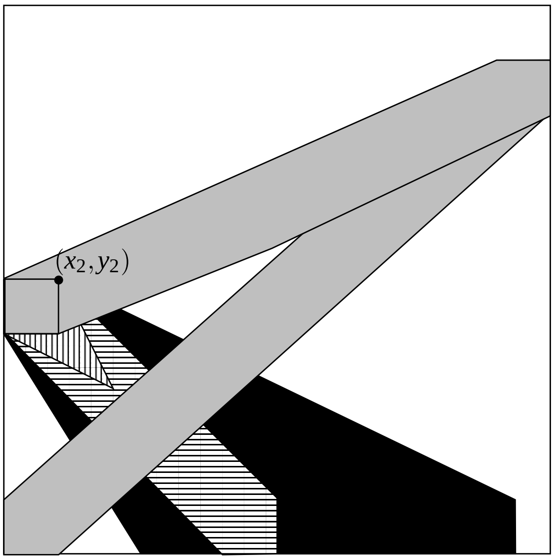

Fig. 3. Comparison of area coverage control algorithms at $t = t_2$.

3.1 Receding Horizon Strategy

The *Receding Horizon* strategy maximizes the new area covered by the vehicle between turn k and $k+1$ neglecting the influence of the error. The feedback control is given by

$$\theta_k^{\mathrm{N}} = \arg \max_{\theta \in [-\pi, \pi)} A(B(x_k, y_k, \theta) \cup C_k),$$

where

$$B(x,y,\theta) = \bigcup_{z \in \ell(x,y,\theta) : \bar{c}(z) \subset \Omega} \bar{c}(z)$$

denotes the set to be covered till next turn if the error was zero. Here $\bar{c}(z)$ denotes the set covered by a vehicle in position $z \in \Omega$ and $\ell : \mathbb{R}^2 \times [-\pi, \pi) \mapsto \mathbb{R}^2$ is the line $\ell(x,y,\theta) = \{(x + s\cos\theta, y + s\sin\theta) : s \geq 0\}$. The union of the striped areas corresponds $B(x_2, y_2, -\pi/4)$ in Figure 3.

3.2 Robust Receding Horizon Strategy

The *RRH* control strategy maximizes the new area that is guaranteed to be covered by the vehicle between turn k and $k+1$. This feedback control law is given by

$$\theta_k^{\mathrm{G}} = \arg \max_{\theta \in [-\pi,\pi)} A(B_{\cap}(x_k, y_k, \theta) \cup C_k),$$

where

$$B_{\cap}(x,y,\theta) = \bigcap_{\alpha \in [-\varepsilon,\varepsilon]} B(x,y,\theta+\alpha)$$

denotes the set guaranteed to be covered regardless of the actual error executed at t_k. In Figure 3, the vertically striped area corresponds to $B_{\cap}(x_2, y_2, -\pi/4)$. Obviously, the area guaranteed to be covered is smaller than the nominal area. Note that this algorithm will not work for large ε. In that case, when a sufficiently large part of the area has been covered at time t_k, say, the guaranteed new area to cover is equal to zero, i.e., $A(B_{\cap}(x_k, y_k, \theta_k) \cup C_k) = A(C_k)$. (When this happens in our implementation, a random control action is issued.)

3.3 Opportunistic Receding Horizon Strategy

The *ORH* control strategy maximizes the area that corresponds to the nominal control but evaluated over the union of all possible errors less than ε. This feedback control law is given by

$$\theta_k^{\mathrm{P}} = \arg \max_{\theta \in [-\pi,\pi)} A(B_{\cup}(x_k, y_k, \theta) \cup C_k),$$

where

$$B_{\cup}(x,y,\theta) = \bigcup_{\alpha \in [-\varepsilon,\varepsilon]} B(x,y,\theta+\alpha).$$

The union of the black and the stripped areas in Figure 3 corresponds to $B_{\cup}(x_2, y_2, -\pi/4)$. The area of this region is larger than the corresponding regions of both preceding algorithms.

3.4 Modified Boustrophedon-Path Strategy

The *Boustrophedon* control strategy mimics a boustrophedon path, which is the simple back-and-forth motion an ox follows when dragging a plow in a field (Choset, 2001). Note that if we would not have error on the direction commands θ this boustrophedon path would give the optimal solution to the optimization problem presented at the beginning of this section. The only difference here is that the Boustrophedon control θ_k^{H}

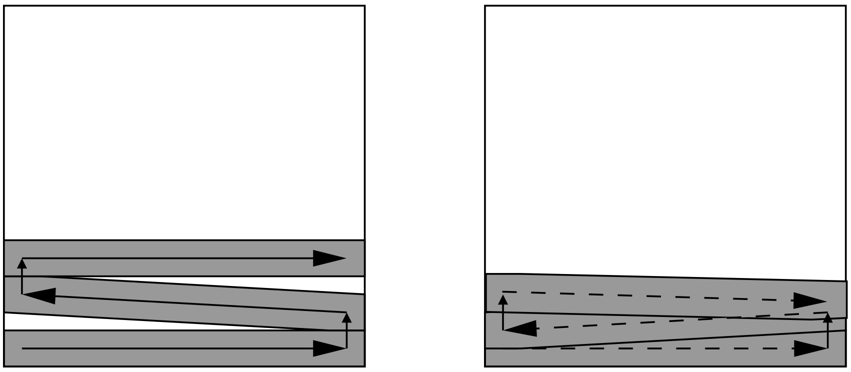

Fig. 4. The left picture shows that a boustrophedon path does not succeed to cover the set Ω, when there is a non-zero steering error ε. The proposed Boustrophedon strategy, however, performs conservative movements to guarantee complete coverage, as shown in the right picture.

is chosen conservatively, so that $C(t)$ is guaranteed to be a connected set for all $t \geq 0$, see Figure 4. Note that a pure boustrophedon strategy, without the error compensation, might not succeed in covering the whole set Ω. For small ε, the Boustrophedon control strategy is efficient in the sense that N is close to optimal. When ε grows, however, the strategy rapidly deteriorates. For ε larger than $\varepsilon_c = 2^{-1} \arctan L^{-1}$ it might happen that the path gets into a closed orbit, and thus does not contribute to the area coverage.

3.5 Randomized Strategy

The *randomized* control strategy is simply to let θ_k^{R} take a random value from the uniform distribution $\mathcal{U}(-\pi, \pi)$. This algorithm is easy to implement, since no state information, such as current position or covered area, is needed. The Electrolux automatic cleaning robot Trilobite (Electrolux, 2001) and the Husqvarna automatic lawn mover Solar Mover (AB, Husqvarna, 1998) apply similar randomized navigation schemes.

4 Complexity of Area Computations

In this section we will discuss the complexity of the required geometrical operations, namely, area, intersection, and union of polygons, for the computation of the algorithms proposed.

The greedy algorithms require information on the global situation, which in practical terms means calculating the area covered by the polygons generated from the vehicle movements.

The Direct Approach is when we compute the area covered after n turns, and the Indirect Approach is when we compute the area remaining to be covered. We will prove that for the Direct Approach the number of operations at iteration n required is $R_D = 2^n$, while for the Indirect Approach the number at iteration n are bounded as $R_I \leq (Aa^n + Bb^n + 1) + 2(Aa^{n-1} + Bb^{n-1})$, with $\#P^c(n) = Aa^n + Bb^n$; $A = \frac{\sqrt{5}+3}{2\sqrt{5}}, B = \frac{\sqrt{5}-3}{2\sqrt{5}}$; $a = \frac{1+\sqrt{5}}{2}, b = \frac{1-\sqrt{5}}{2}$.

4.1 Direct Approach

The Direct Approach computes the covered area by calculating the union of the polygons swept at time n.

A problem arises when using this approach because the union of several convex polygons in general leads to non-convex polygons. Moreover when we have more than three polygons it leads in general to polygons with holes. Having non-convex polygons is not a big issue, but having polygons with holes increases the complexity of computing the area considerably. To avoid this problem we rewrite the area of the union as a function of areas of intersections using de morgan's law. This simplifies the problem as the intersection of two convex polygons leads to another convex polygon.

By induction we get the following expression, with $A(p)$ denoting the area of the polygon p, and p_i referring to the convex polygon generated by the system between instants $i-1$ and i.

$$\begin{aligned} A\left(\bigcup_{i=1}^{n} p_i\right) = & \sum_{i=1}^{n} A(p_i) - \sum_{1 \leq i_1 < i_2 \leq n} A(p_{i_1} \cap p_{i_2}) + \dots \\ & + \sum_{1 \leq i_1 < i_2 < \cdots < i_r \leq n} (-1)^{r+1} A(p_{i_1} \cap \cdots \cap p_{i_r}) \\ & + \cdots + (-1)^{n+1} A(p_1 \cap p_2 \cap \cdots \cap p_n). \end{aligned} \tag{4}$$

The number of terms on the right-hand-side corresponds to the number of area computations. Note that in each sum we have as many terms as ordered combinations of the n polygons taken in groups of r, i.e. $\binom{n}{r}$. Thus the number of area computations at turn n is $\sum_{i=1}^{n} \binom{n}{i} = 2^n - 1$. It can be reduced in practice to only 2^{n-1} by storing the areas computed at turn $n-1$. Similarly the number of required intersections at turn n are given by

$\sum_{i=1}^{n}\binom{n}{i} = 2^n - 1$, which, using a tree-like structure storing the polygons resulting of k polygons intersection to calculate the intersections of $k+1$ polygons, reduces to 2^{n-1}.

Hence the number of geometric computations needed in each iteration for the direct approach is equal to $R_D = 2^{n-1} + 2^{n-1} = 2^n$.

4.2 Indirect Approach

The Indirect Approach to compute the covered area, is based in the se Morgan law. Denote the complement polygon of p_i by $p_i^c = \Omega - p_i$.Then,

$$A\left(\bigcup_{i=1}^{n} p_i\right) = A\left(\Omega - \bigcap_{i=1}^{n} p_i^c\right) = A(\Omega) - A\left(\bigcap_{i=1}^{n} p_i^c\right) \tag{5}$$

Let the number of polygons in $\bigcup_{i=1}^{n} p_i^c$ at iteration n be denoted by $P^+(N)$. It can be shown that $P^+(n)$ is bounded by a Fibonacci-like series

$$\begin{aligned} P^+(n) = P^+(n-1) + P^+(n-2); \\ P^+(0) = 1, P^+(1) = 2 \end{aligned} \tag{6}$$

which has the solution

$$\begin{aligned} P^+(n) = Aa^n + Bb^n; \\ A = \frac{\sqrt{5}+3}{2\sqrt{5}}, B = \frac{\sqrt{5}-3}{2\sqrt{5}}; \\ a = \frac{1+\sqrt{5}}{2}, b = \frac{1-\sqrt{5}}{2} \end{aligned} \tag{7}$$

Hence the number of geometric computations for the indirect method at time n using the Indirect Approach is bounded by $R_I \leq (Aa^n + Bb^n + 1) + 2(Aa^{n-1} + Bb^{n-1})$.

4.3 Comparison

In computing R_D and R_I, it is obviously so that for $n > 4$ R_I is smaller than R_D. Note that for small n the bound of R_I is tight. For large n, the bound is conservative. Thus, the indirect approach is in general doing much better than the direct approach. This can be clearly observed in Figure 5, which has a semilog scale. Notice that R_I is approximated by a line: $R_I \sim \alpha c^{n-1}$, with $\alpha = 5$ and $c = 1.6125$.

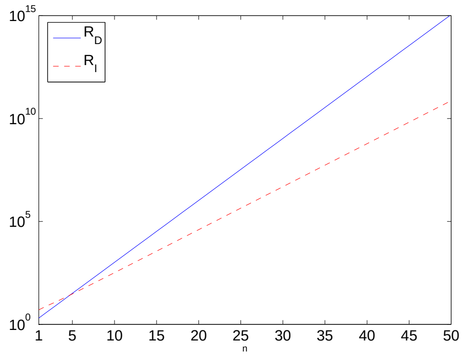

Fig. 5. Comparison of the complexity of computing the covered area through a Direct Approach (R_D) and Indireact Approach (R_I).

5 Simulations

5.1 Implementation

Several controllers have been presented so far for the coverage problem under uncertainty. These algorithms of control can be clearly divided in a group of greedy algorithms, and a Random and an Boustrophedon one. The implementation of a Random strategy or the Boustrophedon strategy presented are straight forward, needing only in the latter case certain feedback on the angle followed by the robot after a straight movement to decide the new turn angle.

All the algorithms were implemented in Matlab. Using Vatti's algorithms for polygon clipping (Vatti, 1992) implementation by Murta in the General Polygon Clipping Library (Murta, 2003), a set of Matlab MEX-files were created to perform the intersection and union of two polygons. Functions for area calculation and convex hull generation by Pankratov (Pankratov, 1995) were also used after slight modifications for compatibility with the latest versions of Matlab.

Convex hull generation functions were used to generate the polygon swept by the robot in a movement taking as input the corner points of the robots surface in the initial and final positions in the border of Ω. Using the polygon clipping algorithms the polygons covered by the robot are actualized with the new covered polygon, and then the area covered in this new situation is calculated using Pankratov's area calculation algorithm. This process is done after a command is issued to update the coverage map but also for each direction θ before performing a movement of the robot, and according to the nominal coverage strategy the robot will be sent in the direction that shows a bigger coverage. In the cases of ORH and RRH coverage slight modifications are needed. In the ORH coverage strategy, the polygon in the θ tested direction was generated giving as input to the convex hull generation function the corner points of the robot in its initial position, and the two final positions in the border of Ω for the directions $\theta + \varepsilon$ and $\theta - \varepsilon$. In the RRH case two different polygons were generated first as in the Nominal case, for the directions $\theta + \varepsilon$ and $\theta - \varepsilon$, and then intersected to generate the guaranteed coverage polygon that was used as optimization area in the following steps of the algorithm.

In all the algorithms the area calculations were implemented using the Indirect Approach presented in 4.

5.2 Evaluation and Results

To evaluate the area coverage control strategies, the results from Monte Carlo simulations are presented in this section. The size of the set Ω to be covered is set to $L = 10$. The turning error e_k, $k = 0, \ldots, N$, is drawn from a uniform distribution $\mathcal{U}(-\varepsilon, \varepsilon)$ (except for the last part of the section, where a comparison with normally distributed errors is made).

Figure 6 shows a snapshot of a simulation with $\varepsilon = 0.078$ at $t = t_4$. The upper left plot shows the accumulated covered set $C(t_4)$ in white, with the current coverage $c(t_4)$ marked by a small square. At this stage the covered area is equal to $A(C(t_4)) \approx 46\%$. Recall that the vehicle starts in the lower left corner $(x(0), y(0)) = (1, 1)$. Note that the error e_0 leads to that the vehicle is not able to steer exactly to the upper right corner. The upper right plot indicates the estimation for the nominal control, while the lower left and the lower right shows the RRH and ORH controls, respectively. All the actions in this simulation are taken based on the nominal controller.

Figure 7 shows a snapshot of the same simulation at $t = t_{25}$. At this much later state of the simulation almost all of Ω is covered, namely, $A(C(t_{25})) \approx 98\%$.

An extensive simulation comparison of the five area cover control strategies is shown in Figures 8 and 9. For each value of the error bound ε marked in the figure, one hundred Monte Carlo simulations were done and the average N was derived for 98% coverage. The ε-axis can roughly be divided into four regions. For small errors ($\varepsilon < 0.07$), the Boustrophedon control strategy gives the best result. Note that for $\varepsilon = 0$, it gives $N = 18$, which is the optimal. For $\varepsilon > 0.07$, the strategy shows quickly bad performance. This is related to that $\varepsilon_c = 0.05 < \varepsilon$ in this case, see Section 3. For $\varepsilon \in (0.07, 0.10)$, the nominal and the RRH control strategies are equally good. Then for

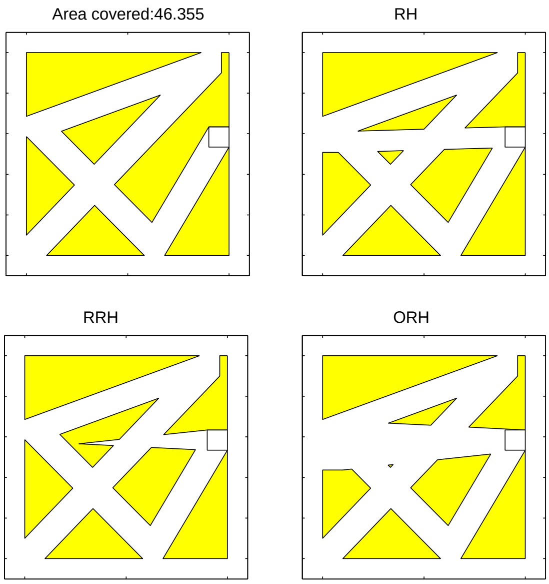

Fig. 6. Snapshot of an area coverage simulation at turn $t = t_4$ using a nominal controller to decide the actions. About half of the area is covered, as indicated by the white part of Ω. The plots also illustrates the nominal, RRH, and ORH control strategies.

$\varepsilon \in (0.1, 0.4)$, the nominal control is the best. For large errors ($\varepsilon > 0.4$), the randomized strategy perform similarly, which is natural because the worst-case error is then larger than 23 degrees. For a given vehicle model, Figures 8 and 9 indicate hence preferable choices of feedback controls. Though it should be emphasized that the implementation complexity varies for the different control strategies.

The nominal control strategy shows a quite good performance over a large range of error bounds. Figure 10 shows the same result as in Figure 8 for this algorithm, but includes the standard deviations.

Note that there are some abnormalities in Figure 8, in particular the peak of the RRH strategy. To explain this peak we have to first explain why we are obtaining results worse than a randomized strategy for both the RRH and ORH strategies. This worse results are a product of what we denoted as ”border trapping effect”. This effect is given by the fact that once there is a certain amount of the area covered, the regions uncovered start to concentrate on the borders of Ω. And thus, in the case of the ORH and RRH strategy forcing them to decide as best direction to follow the one parallel to the border. If we add to that command the turning error, we will obtain that in average half of

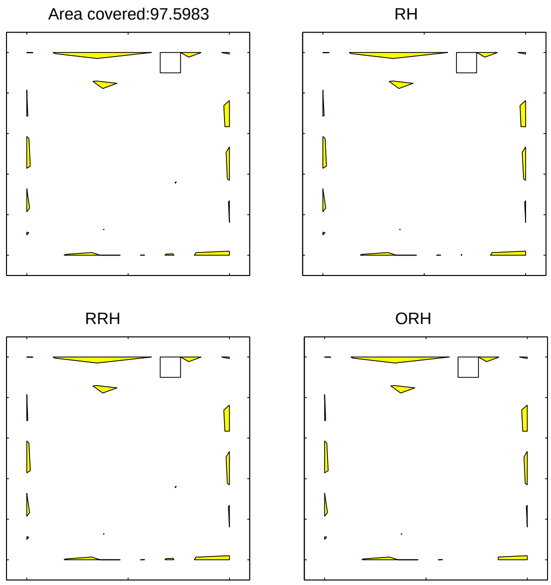

Fig. 7. Snapshot of the area coverage simulation from Figure 6 at $t = t_{25}$. Almost all of the area is covered.

our turns will move the robot against the wall, and thus not moving from its current position and not contributing to the coverage. Now, going back to the explanation of the peak in the RRH strategy, we can notice how after that peak the results of this strategy converge rapidly toward the randomized ones. In the current implementation when the RRH strategy obtains no additional guaranteed coverage in any of the directions but still is aware that we have not achieved complete coverage, randomized commands are issued, explaining the afore mentioned convergence.

It is interesting to see how influential the error distribution is on the results. Figure 11 shows a comparison between errors e_k from the uniform distribution $\mathcal{U}(-\varepsilon,\varepsilon)$ (marked with rings) and errors from the normal distribution $\mathcal{N}(0,\varepsilon/\sqrt{3})$ (asterisks). The distributions have the same means and standard deviations. As expected, the normal distribution yields a slightly lower N, but still the results are comparable.

6 Experiments

The experimental setup, see Figure 12, is based on the Khepera II mobile robot (K-Team, 2002), see Figure 13. The mobile robot Khepera II is a small self-

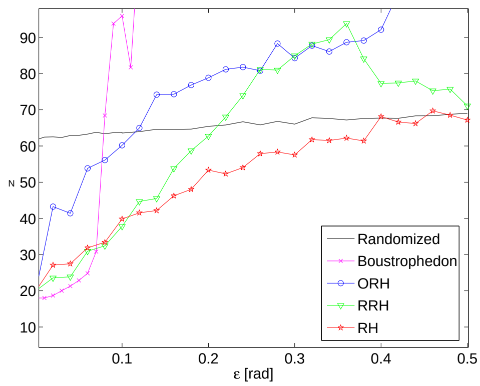

Fig. 8. The average number of turns N required for 98 % coverage versus error bound ε. Five control strategies are compared. Each mark corresponds to one hundred Monte Carlo simulations. The control strategy that gives the best performance depends on ε.

contained wheeled robot complete with processor and basic sensors (infra-red proximity sensors and encoders). Its main characteristics are a diameter of 70 mm (around 55 mm for the top surface), a precise odometer and a linear speed in the range 0.02–1.00 m/s. The Khepera II infrared proximity and ambient light sensors have a range of up to 100 mm and the high precision encoders in each wheel are capable of a resolution of 12 pulses per millimeter. The region to be covered in the experiments has a side of $L = 550$ mm, so the experimental setup and the simulation study have roughly the same quota $A(c)/A(\Omega)$.

The experimental setup is completed with a test arena (region to be covered), with an overhead camera (LEGO-cam) connected to a computer (Pentium III). The Khepera II robot is also connected to the computer where the algorithms of control are running on Matlab. Some image processing algorithms, also in Matlab, together with the proximity sensors information are used for the localization of the robot at the boundary of Ω and thus give feedback information to the control algorithms.

Note that even if the coverage algorithm is developed for a robot covering a square, it is easily modified to deal with circular robots. Obviously, we may also make a conservative approximation by considering the largest square covered by the corresponding disc.

To obtain a good bound ε of the error present in our system, some experiments were performed. The experiments performed were a set of turns with different command angles θ_k, that were compared with the angles $\overline{\theta}_k$ read from the computer vision based

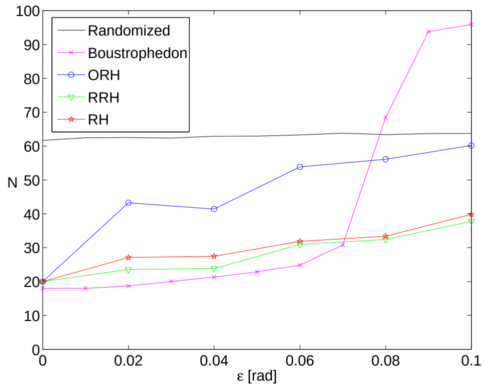

Fig. 9. Detail of Figure 8 in the range of small error bound ε

localization system to obtain the turning error $e_k = \left|\bar{\theta}_k - \theta_k\right|$. The system was run for a sequence of turns without resetting the system to an original position so that this bound would also include the accumulative effect of the error in the designed system. After these experiments, the mean value of the error $\bar{e}$ and its standard deviation σ_e were calculated and the ε that was used for the whole system thereafter set to $\varepsilon = \bar{e} + \sigma_e$ obtaining a value of $\varepsilon = 0.078$.

As illustrated by the snapshot in Figure 14, the experiment, using a nominal controller, follows the behavior of the corresponding simulations (Figure 6) quite well. When running an experiment for a long time, it has been noticed however that the error model used in the simulation is not accurate. The error distribution tends to change over time. This is particularly the case if the localization error at the boundary of Ω is not negligible.

7 Conclusions

Motivated by the need for robust control algorithms for area coverage under uncertain vehicle models, we presented and analyzed a few possible strategies. It was shown that the number of turns needed in order to cover an area is increasing with the error bound ε of the turns. Moreover, which algorithm that performed the best depends on ε. For example, for a bad steering actuator (large ε), a randomized algorithm performed as well as the more intelligent ones, while for a better actuator considerable improvements can be achieved by using the proposed robust strategies.

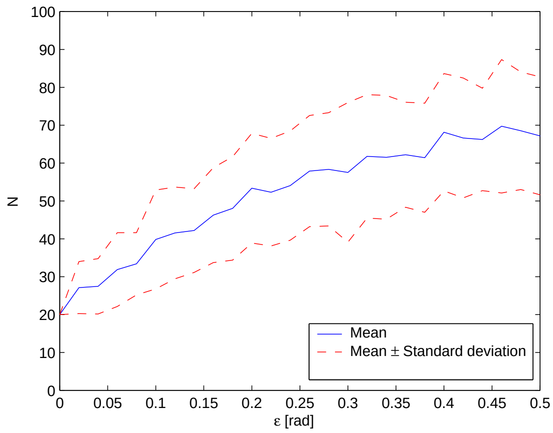

Fig. 10. Similar simulations as in Figure 8 for the nominal control strategy. The dashed curves indicate the standard deviation.

The closed-loop control system for the area coverage was presented as a hybrid automaton. In this way, it was possible to have a low-order model that still can capture the complexity of the problem. It would be interesting to apply existing verification tools in order to analyze this so called timed automata (Alur & Dill, 1994), which the area coverage problem in the paper led to. Another possible extension of the work is to consider collaborating vehicles.

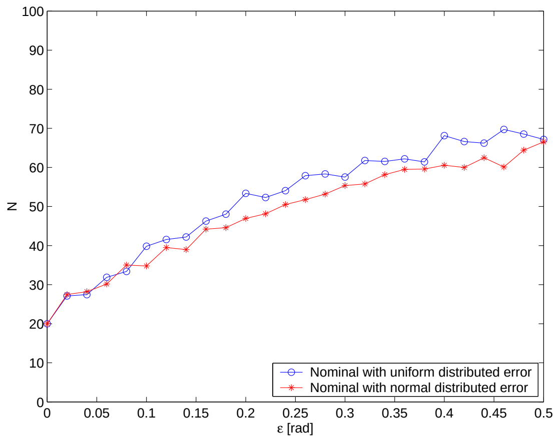

Fig. 11. Similar simulations as in Figure 8 for the nominal control strategy. The rings indicate the results for uniformly distributed errors, while the asterisks indicate normally distributed errors.

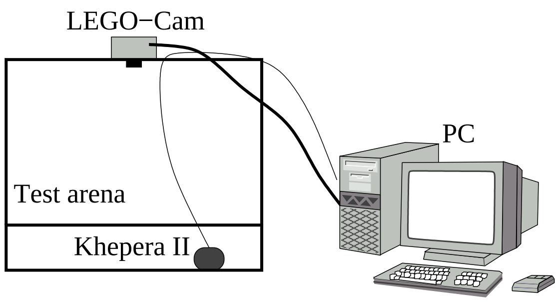

Fig. 12. Experimental setup sketch showing the computer where the algorithms are running, the robot and the test arena with the overhead camera.

Fig. 13. Experimental setup for evaluation of the area coverage robot.

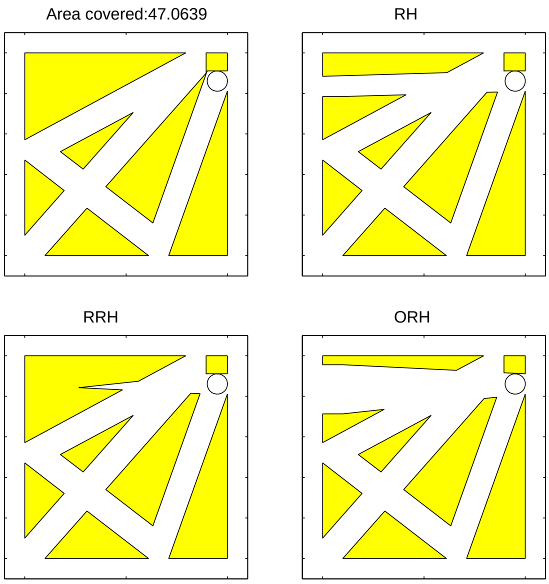

Fig. 14. Snapshot of an area coverage experiment at turn $k = 4$ using the nominal control strategy. The experimental results agrees well with the simulations.

References

AB, Husqvarna. 1998 (April). *Husqvarna Solar Mower*. `http://www.solarmower.com`.

Alur, R., & Dill, D. L. 1994. A theory of timed automata. *Theoretical Computer Science*, **126**(2), 183–235.

Balch, T., & Arkin, R. C. 1995. Communication in reactive multiangent robotic system. *Autonomous Robots*, **1**(1).

Balluchi, A., & Souères, P. 1996 (December). Optimal Feedback Control of Dubins' Car Tracking Circular Reference Paths. *Pages 3558–3563 of: Proc. 35th IEEE Conference on Decision and Control*.

Choset, H. 2000. Coverage of Known Spaces: The Boustrophedon Cellular Decomposition. *Autonomous Robots*, **9**, 247–253.

Choset, H. 2001. Coverage for robotics - A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 113–126.

Choset, H., & Pignon, P. 1997 (December). Coverage path planning: The boustrophedon decomposition. *In: Proceedings of the International Conference on Field and Service Robotics, Camberra, Australia*. IEEE.

Choset, H., Acar, E., Rizzi, A., & Luntz, J. 2000. Exact cellular decompositions in terms of critical points of Morse functions. *In: IEEE International Conference on Robotics and Automation*.

Colegrave, J., & Branch, A. 1994. A case study of autonomous household vacuum cleaner. *In: AIAA/NASA CIRFFSS*.

Electrolux. 2001. *Trilobite*. `http://trilobite.electrolux.com/`.

Gonzalez-Banos, H.H., & Latombe, J.C. 2001. A Randomized Art-Gallery Algorithm for Sensor Placement. *Pages 232–240 of: Proc. 17th ACM Symp. on Computational Geometry*. ACM.

Greiner, G., & Hormann, K. 1998. Efficient Clipping of Arbitrary Polygons. *ACM Transactions on Graphics*, **17**(2), 71–83.

Hert, S., Tiwari, S., & V.Lumelsky. 1996. A terrain-covering algorithm for an AUV. *Autonom. Robots*, **3**, 91–119.

Huang, W. H. 2001. Optimal Line-sweep-based Decompositions for Coverage Algorithms. *In: IEEE International Conference on Robotics and Automation*.

Huang, Y.Y., Cao, Z.L., & Hall, E.L. 1986. Region filling operations for mobile robot using computer graphics. *Pages 1607–1614 of: Proceedings of the IEEE Conference on Robotics ans Automation*.

K-Team. 2002. *Khepera II*. `http://www.k-team.com/`.

Kurabayashi, D., Ota, J., Arai, T., & Yoshida, E. 1996. Cooperative sweeping by multiple mobile robots. *In: International Conference on Robotics and Automation*.

Latombe, J. C. 1991. *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers.

MacKenzie, D., & Balch, T. 1996. Making a clean sweep: Behaviour based vacuuming. *In: AAAI Fall Symposium*.

Murta, A. 2003 (Jan). *A General Polygon Library*. `http://www.cs.man.ac.uk/aig/staff/alan/software/`.

Ollis, M., & Stenz, A. 1996. First results in vision-based crop line tracking. *In: International Conference on Robotics and Automation*.

Oommen, B. J., Iyengar, S. S., Rao, N. S. V., & Kashyap, R. L. 1987. Robot Navigation in Unknown Terrains Using Learned Visibility Graphs Part I: The Disjoint Convex Obstacle Case. *IEEE Journal of Robotics and Automation*, **3**(6), 672–681.

Pankratov, K. K. 1995. *SAGA Matlab Toolbox*. `http://puddle.mit.edu/~glenn/kirill/saga.html`.

Poduri, S., & Sukhatme, G. S. 2004. Constrained Coverage for Mobile Sensor Networks. *In: IEEE International Conference on Robotics and Automation*.

Preparata, F.P., & Shamos, M.I. 1985. *Computational Geometry: An Introduction*. Springer-Verlag. Pages 198–257.

Rao, N. S. V., & Iyengar, S. S. 1990. Autonomous Robot Navigation in Unknown Terrains: Incidental Learning and Environmental Exploration. *IEEE Transactions on System, Man and Cybernetics*, **20**(6), 1443–1449.

Svennebring, J., & Koenig, S. 2004. Building Terrain-Covering Ant Robots: A Feasibility Study. *Autonomous Robots*, **16**, 313–332.

Vatti, B. R. 1992. A Generic Solution to Polygon Clipping. *Communications of the ACM*, **35**(7), 56–63.

Zelinsky, A., Jarvis, R. A., Byrne, J. C., & Yuta, S. 1993. Planning paths of complete coverage of an unstructured environment by a mobile robot. *Pages 533–538 of: Proceedings of International Conference on Advanced Robotics*.