INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center Street • Suite 600 • Berkeley, California 94704 • 1-510-642-4274 • FAX 1-510-643-7684

Navigation Without Perception of Coordinates and Distances

Armin Hemmerling*

TR-93-018

March 1993

Abstract

We consider the target–reaching problem in plane scenes for a point robot which has a tactile sensor and can locate the target ray. It might have a compass, too, but it is not able to perceive the coordinates of its position nor to measure distances. The complexity of an algorithm is measured by the number of straight moves until reaching the target, as a function of the number of vertices of the (polygonal) scene.

It is shown how the target point can be reached by exhaustive search without using a compass, with the complexity $\exp(O(n^2))$. Using a compass, there is a target–reaching algorithm, based on rotation counting, with the complexity $O(n^2)$.

The decision problem, to recognize if the target cannot be reached because it belongs to an obstacle, cannot be solved by our type of robot. If the behaviour of a robot without compass is periodic in a homogeneous environment, it cannot solve the target–reaching problem.

*Ernst Moritz Arndt University Greifswald, Department of Mathematics and Computer Science, D-O-2200 Greifswald, Jahnstr. 15 a, Germany (hemmerling@math-inf.uni-greifswald.dbp.de). Parts of this work were done while the author was visiting the ICSI.

1 Introduction

Problems of algorithmic motion planning or navigation in environments with obstacles have been studied in several variants and contexts. Especially this has been done within the framework of computational geometry, for surveys see [Y] and [HA]. In most of these papers the off–line point of view is taken, i.e. the robots completely know (a map of) the scene in which they have to work.

During the last five years, the interest in on–line algorithms of motion planning has grown, as the papers [BBFY] , [BRS] , [K] , [LS] , [PY] and their references show. Here the robot knows only that part of the scene which has been perceptible up to the current step. But it is usually supposed that it can perceive the coordinates of its current position, sometimes also of the visible parts of the obstacles.

What is a minimum equipment which enables a point robot to reach a fixed target point in a plane scene with finitely many polygonal obstacles, whenever this is topologically possible? At least, it should have a tactile sensor such that it can notice that it touches some obstacle and can follow the boundary of a touched obstacle.

We will show how the task can be solved by exhaustive search if, in addition, the robot is able to locate and to follow (as long as it does not meet an obstacle) the target ray connecting the current position with the target. To perform our exhaustive search, the robot needs the computational ability to generate a *universal* infinite sequence of move instructions from a suitable three–letter alphabet. This is a sequence containing every finite word as a substring. So it is extremely non–periodic. On the other hand, we know that a robot without compass with a periodic behaviour in a homogeneous environment cannot solve the target–reaching problem. The worst case complexity of the exhaustive search is $\exp(O(n^2))$.

If, moreover, the robot is equipped with a compass allowing to measure the angle between the target ray and the positive x–axis, and the angle of any currently touched edge of an obstacle, then there is a target–finding procedure with the complexity $O(n^2)$. That is based on the method of rotation counting. But even such a robot is not able to recognize if the task is topologically unsolvable, i.e. if the target is included in some obstacle.

The *size* of a scene is the number n of vertices of obstracles. Depending on this size, the complexity measure used here counts the number of steps performed by the robot until the target is reached. One step corresponds to a move between consecutive *breakpoints* of the searching procedure, which are caused either by reaching some obstacle in following the target ray, or by reaching a vertex of the scene in following the boundary of a touched obstacle.

In contrast to on–line motion planning procedures in computational geometry, our algorithms don't suppose that the robot is able to perceive the coordinates of its position or to measure distances of points or lengths of paths in the scene. We will even see that the task becomes nearly trivial under those suppositions. Therefore, our point of view corresponds more to labyrinth theory than to computational geometry. Indeed, most of the techniques used in the present paper come from the automaton–oriented labyrinth research presented in the monograph [H]. Especially the method of rotation counting (regular swinging) is essentially used in [A] already, where it really shows the power of the compass, whereas the search procedure in [BK] is also based on rudiments of length measuring.

The present paper is organized as follows. After introducing the basic notations in Section 2, we show how the problems can easily be solved by a robot which perceives the coordinates of its position. In Section 3, the unsolvability of the decision problem by our restricted type of automaton is proved. Moreover, we will see that a robot without compass cannot solve the target–reaching problem if it behaves periodically in a homogeneous environment. Section 4 descibes the solution of this problem via exhaustive search, without using a compass. Sections 5 and 6 presents the main results of the paper. They lead to an efficient target–reaching algorithm. We close with a discussion of the results and open problems.

2 Basic Concepts

By a *finite family of obstacles*, we understand a finite sequence

$$\mathcal{P} = (p_i : 1 \leq i \leq m),$$

$m \in \mathbf{N}$, of oriented, simply closed polygonal paths p_i in the Euclidean plane $\mathbf{R}^2$, such that the interior of every p_i lies on the left–hand side of the path. Every p_i is determined by a finite sequence $(P_1, \ldots, P_{k_i}, P_{k_i+1} = P_1), k_i \geq 3$, of *vertices* $P_j, 1 \leq j \leq k_i$, where the *edges* (or *sides*) $\overline{P_j P_{j+1}}$, for $1 \leq j \leq k_i$, have no further common points. More precisely, if $\overline{P_{j_1} P_{j_1+1}} \cap \overline{P_{j_2} P_{j_2+1}} \neq \emptyset$ for $1 \leq j_1 < j_2 \leq k_i$, then $j_2 = j_1 + 1$, and P_{j_2} is the only common point of these straight line segments. Recall that, by Jordan's curve theorem, the interior of p_i is a simply connected, open region in the plane, and p_i is its oriented boundary. In the sequel, we will sometimes identify p_i with its carrier set $\bigcup_{j=1}^{k_i} \overline{P_j P_{j+1}}$. Let $\mathcal{O}_i$ denote the interior of p_i, it will be referred to as the i–th *obstacle*.

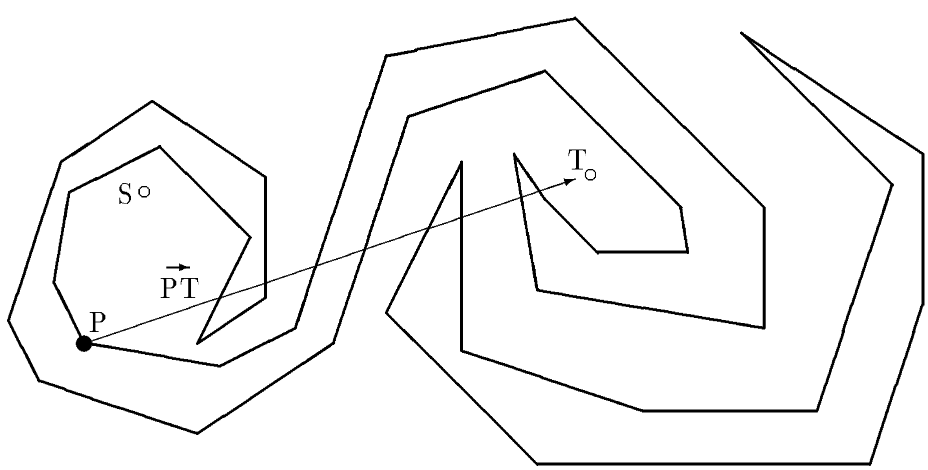
We will suppose that the closures $\mathcal{O}_i \cup p_i$, for $1 \leq i \leq m$, are pairwise disjoint, i.e. the obstacles do neither join nor touch each other.

A *scene* is a triple

$$\mathcal{S} = (\mathcal{P}, T, S),$$

where $\mathcal{P}$ is a finite family of obstacles, $T \in \mathbf{R}^2$ is the *target*, and $S \in \mathbf{R}^2 \setminus \bigcup_{i=1}^{m} \mathcal{O}_i$, $S \neq T$, is the *starting point*, see Figure 1 for an illustration.

Figure 1:



Our basic task is to establish (efficient) algorithms by which an autonomous robot, starting from the point S in an arbitrary scene $\mathcal{S}$, is able to reach the target T, whenever this is topologocally possible, i.e. if $T \notin \bigcup_{i=1}^{m} \mathcal{O}_i$. Besides this *target–reaching problem*, the *decision problem* will be considered, too. Here, in every scene $\mathcal{S}$, the robot has to halt after some finite time with the result "target not reachable" (in a suitably encoded form, maybe by the internal state) if and only if the target–reaching task cannot be solved because $T \in \bigcup_{i=1}^{m} \mathcal{O}_i$.

The robot is geometrically represented by a point. We suppose that it is equipped with a tactile sensor which detects if it reaches the target T or the boundary of some obstacle. It is not able to perceive the coordinates of its position, but, in any position $P \neq T$, it can locate the *target ray* $\overrightarrow{PT}$, this is the *ray* starting in P and going through T.

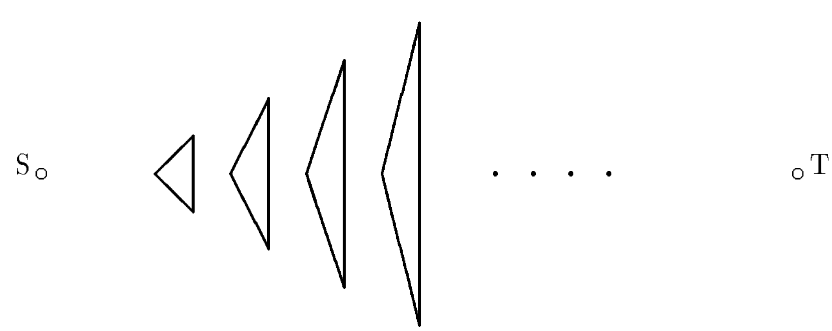
Correspondingly, the robot can move along the *straight line* $\overleftrightarrow{PT}$, both in the direction to T and away from T, until reaching T or a boundary of an obstacle, and it can move along the boundary of a touched obstacle, both in the positive and in the negative orientation, up to the next vertex of this boundary. More precisely, the *breakpoints* of the robot's move are given by the moments in which it reaches T or an obstacle in moving along $\overleftrightarrow{PT}$, or in which it reaches a vertex in moving along a boundary. Additionally, let the starting position in S create a breakpoint. In these breakpoints, depending on its internal state and the signals percepted by its sensor, it has to determine the next direction of move which will be straight up to the next breakpoint.

We hope that this informal description sufficietly characterizes the type of robot we want to deal with. So we omit further details. Concerning the internal states, i.e. the storage, and the computational ability, no restriction is made at this time.

We will say that the robot is equipped with a *compass* if it can additionally measure the angle between the positive $x-$axis and the target ray $\overrightarrow{PT}$, in any position P, and the angle of a currently touched edge of an obstacle. Then it should be able to store (finitely many) values of angles and to compare them with the currently measured angles. This means, there are created further breakpoints of move which correspond to those moments in which a percepted angle takes one of the stored values. For example, a robot with a compass can store the angle of the ray $\overrightarrow{ST}$, at the starting point S. This causes a breakpoint of move whenever the current position belongs to the straight line $\overleftrightarrow{ST}$.

Following [BRS], we measure the *size* of a scene by the number of vertices of the obstacles. Therefore, we say that the *complexity* of a target–reaching or decision algorithm is (bounded by) some function $f : \mathbf{N} \longrightarrow \mathbf{N}$ if, in any scene with just n vertices, it either halts never or halts after at most $f(n)$ steps of working, i.e. after reaching at most $f(n)+1$ breakpoints of move.

Figure 2:

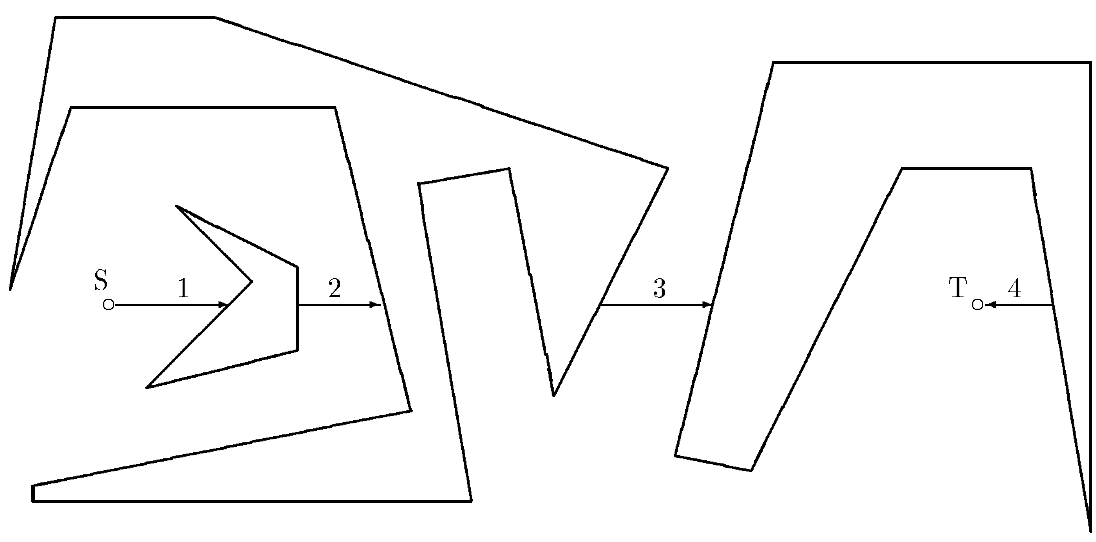


One easily shows that complexity bounds for our target–reaching algorithms must at least linearly increase. Indeed, Figure 2 shows how to construct a scene with n 3–vertex obstacles such that every target–reaching algorithm for a robot with a tactile sensor, target–ray location and compass only, needs more than $2n$ straight line moves from S to T.

As a further interesting illustration, we remark that a robot with a compass and with the additional ability to know the coordinates of its current position (at least in the breakpoints of move), to store some of these values and to compare them with the current ones, is able to solve both the target–reaching and the decision problem in linear complexity $O(n)$. A corresponding algorithm is due to [LS], the correctness and the estimation of complexity can easiliy be verified. We sketch here a slightly modified version, for a scene $\mathcal{S}$. Figure 3 illustrates its work if the target ist reachable:

1. Move straightly in the direction of the target T until reaching T or touching an obstacle.
2. If T is reached, then halt ("target reached").
   If an obstacle ist touched, then go around its boundary p and determine (the coordinates of) that point $P \in p \cap \overleftrightarrow{ST}$ which has the minimal distance to T. Move to P (along p).
3. If the target ray at P points into the touched obstacle, then halt with "target not reachable". (Indeed, in this case the currently touched obstacle includes T.)
   Otherwise, goto 1.

Figure 3:



In [BRS], it is shown how to construct a $(6 \cdot k + 4)$–vertex scene with only one obstacle, for any $k \geq 2$, such that every on–line algorithm (even if it perceives the currently visible part of the scene) needs more than $3 \cdot (k - 2)$ steps up to reaching the target. In [PY] it is proved that the competitive ratio of the number of steps done by an on–line algorithm to the number of steps of a shortest path connecting the starting point with the target cannot be bounded by a constant.

Finally, we remark that every robot which solves the decision problem and percepts obstacles only by a tactile sensor, also solves the target–reaching problem. Indeed, let the robot halt with the result “target not reachable” in every scene where the target belongs to some obstacle. Assume that it does not reach the target in some scene $\mathcal{S}$ where this is topologically possible. Then it does never enter a certain ε–neighbourhood of the target. By adding an obstacle that includes the target and is contained in this ε–neighbourhood, one obtains a scene $\mathcal{S}'$ in which the target is not reachable, but the robot behaves like in the original one, i.e. it does not remark this unreachability.

3 Two Trap Constructions

First we show that robots, which are not able to perceive the coordinates of their position, cannot solve the decision problem.

Theorem 3.1 *There is no robot in the sense of Section 2 (with a tactile sensor, target–ray location, and compass) which solves the decision problem, i.e. which, put in an arbitrary scene, halts with the result "target not reachable" after finitely many steps if and only if the target is topologically not reachable.*

The basic idea of the proof is well–known from labyrinth theory: A compass automaton without the ability of length measuring cannot distinguish between a rectangle and a sufficiently deep rectangular spiral. We will put the target in the centre of the figure. To ensure that the robot does not remark any difference if the object is deformed into a spiral, we start already with the suitably shaped polygon shown in Figure 4 a). Here let the slanted lines belong to the pair of straight lines having the slopes 1 resp. -1 and crossing each other in the target T.

More precisely, we assume that there is a robot of the type described above which solves the decision problem, and we put it into the scene shown in Figure 4 a). If this robot would move away from T on the straight line $\overleftrightarrow{PT}$, in some position P, it would never halt. Therefore, it must move along the boundary of the given obstacle, and it halts after a certain number k of breakpoints with the result "target not reachable".

But then it would behave analogously in the scene sketched in Figure 4 b), where the sloped lines have the same property as obove. Now the obstacle is a sufficiently deep "spiral" such that the robot does not reach an end piece, in k steps. Hence, it would also halt after k breakpoints with the same result "target not reachable", i.e. it would not solve the decision problem. □

Because of Theorem 3.1, we restrict ourselves to the target–reaching problem in the sequel.

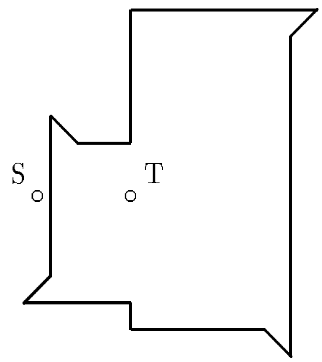
In the remaining part of this section, robots without compass are considered. They have tactile sensors and can locate the target rays only. We will show that such robots are not able to solve the target–reaching problem if their behaviour in certain homogeneous environments becomes periodic. Especially, this holds for finite automata, but also for 1–counter and even for pushdown automata.

The claim for finite automata can be obtained from Budach's result [B] that no finite automaton can escape from all two–dimensional mazes, see also [H]. The self–contained proof of the general result is rather lengthy and will be sketched here only. The reader, who is preferably interested in target–reaching algorithms, should skip to the next section.

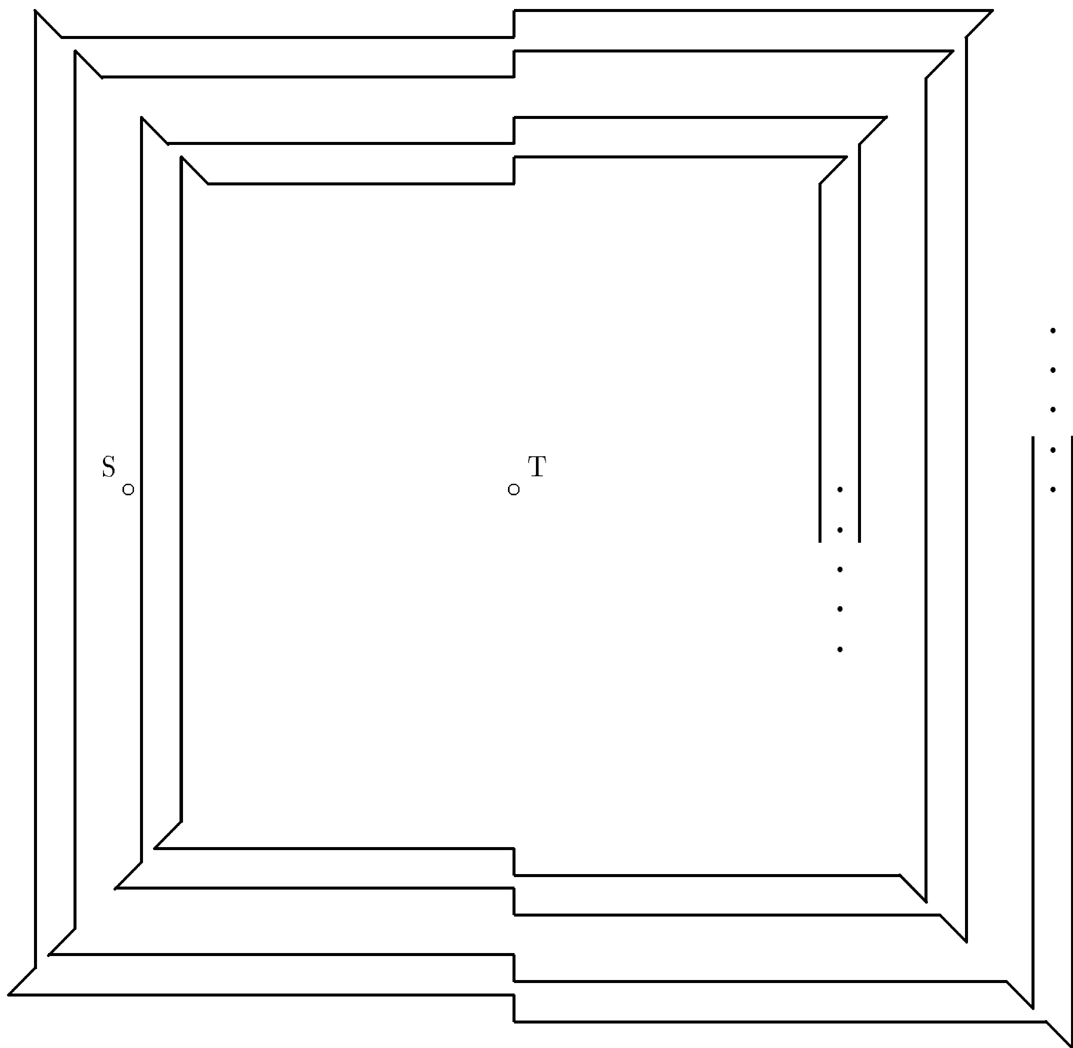
The *homogeneous environments* we will consider here are grids composed of so–called *X–fragments*. These consist of five *corridors* labelled by the letters a, b, c, d, m, as shown in Figure 5 a). By some additional requirements, we secure that the behaviour of a robot without compass does neither depend on the lengths of the corridors nor on the angles in which they join each other. We hope, the meaning of this statement becomes clear in the course of the proof sketch, without a precise definition of the concept of behaviour of a robot.

Figure 4:

a)



b)



More precisely, in composing scenes of X–fragments, we suppose the following conditions.

- The four thick lines in Figure 5 a) represent straight boundary segments parallel to their target rays. This means, they are parts of the straight lines $\overleftrightarrow{PT}$, where T is the target of the scene, and P an arbitrary point of the corresponding boundary segment.
- The thin lines represent straight boundary segments of the scene which are not parallel to the target rays of their points. Let the two sides of the corridors b, m resp. d be parallel to each other.
- The target rays of the two dotted points have the same directions as the corresponding thick line segments on the opposite sides of the corridor m. Especially, no target ray from the corridor c meets the corridor a.
- The maximal width of the corridors is very small compared with the minimal length of the boundary segments of the scene. (For a better presentation, this condition has not been taken into account in drawing Figure 5 a).)

To simplify the drawings and explanations, in the sequel we represent the corridors by thick line segments and label the openings by letters from the alphabet $A = \{a,b,c,d\}$, as shown in Figure 5 b). Furthermore, we also use the simple scheme from Figure 5 c) to represent an arbitrary X–fragment, arranged like in the Figures 5 a), b).

Given a robot with a tactile sensor and target–ray location only, we consider its behaviour within the half-edge–labelled infinite regular grid $\mathcal{G}_X$ consisting of X–fragments which are linked as shown in Figure 5 d). Let the starting point be arbitrarily fixed within the corresponding fragment. Then the path of the robot in $\mathcal{G}_X$ can uniquely be described by a finite or infinite sequence over the alphabet A,

$$f \in A^* \cup A^\omega.$$

f is the sequence of the labels of the openings of the X–fragments through which they are left by the robot in walking in $\mathcal{G}_X$. It is finite if and only if the robot finally remains within some X–fragment.

We will say that the robot's *behaviour in a homogeneous environment is periodic* if the sequence f corresponding to its path in $\mathcal{G}_X$ is finite or finally periodic, i.e. it has the form

$$f = u_0 \cdot u^\omega,$$

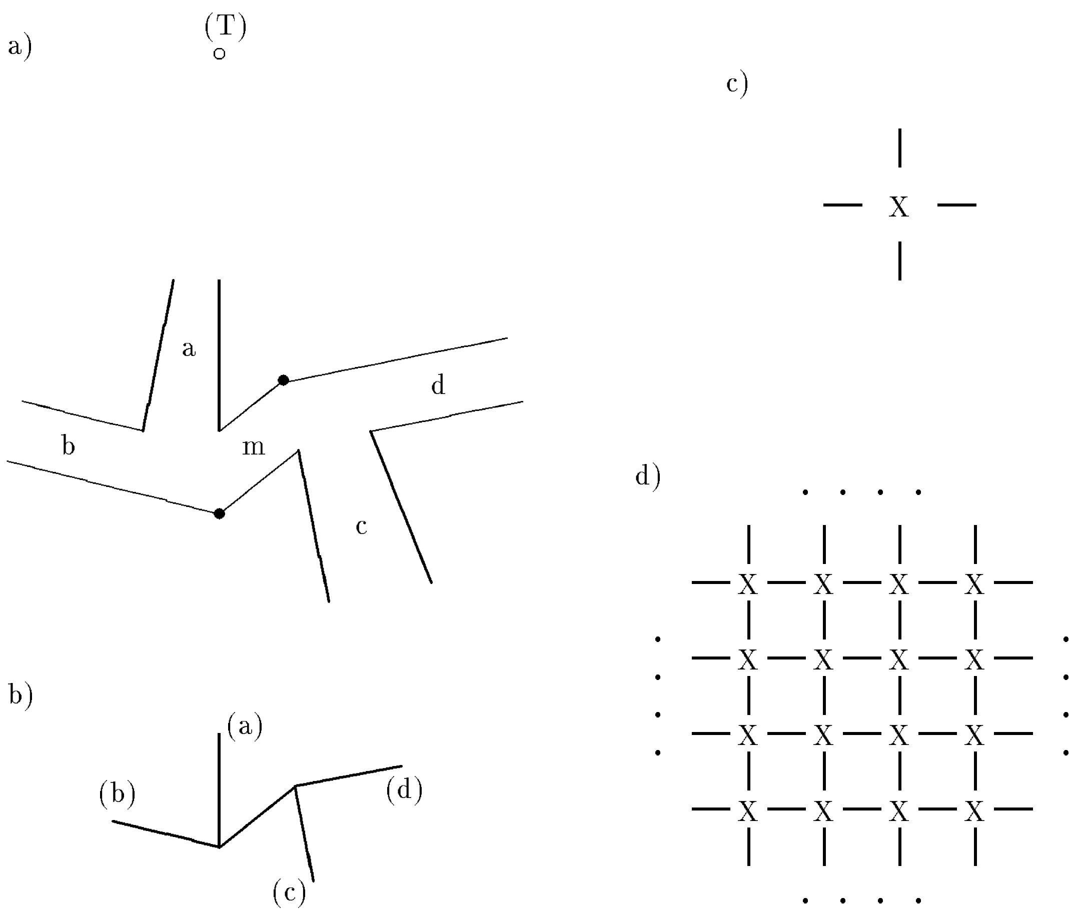
for some words $u_0, u \in A^*$. [1]

For example, if the robot is a finite automaton, it fulfills this condition. But also a 1–counter automaton or even a pushdown automaton with tactile sensor and target–ray location only, have a periodic behaviour in a homogeneous environment.

What does it mean that a robot with tactile sensor and target–ray location is a pushdown automaton? In any breakpoint, with some position P in the scene, the input information

[1] Instead of the grid $\mathcal{G}_X$, one could equivalently use any 4–regular graph composed of X–fragments in such a way that a–corridors are linked with c–corridors, and b– with d–corridors. This would not change the sequence f.

Figure 5:



of the robot says whether $P = T$ (target) or $P \in p_i$ for some boundary p_i, and which direction from P along the straight line $\overleftrightarrow{ST}$ is blocked by an obstacle. This can be coded by a letter from a finite input alphabet. Depending on this input and the internal state, the robot enters a new internal state and chooses a straight move. These reactions can be determined by a state–transition function and an output function. The latter gives the move instruction which can be coded in a finite alphabet, too. So the robot is represented as a discrete, deterministic automaton, and it is straightforward to specify the restrictions for the state set and for the functions in the case of a (deterministic) pushdown automaton. For robots with a compass, this would be more complicated, since the values of angles in polygonal scenes form an infinite set.

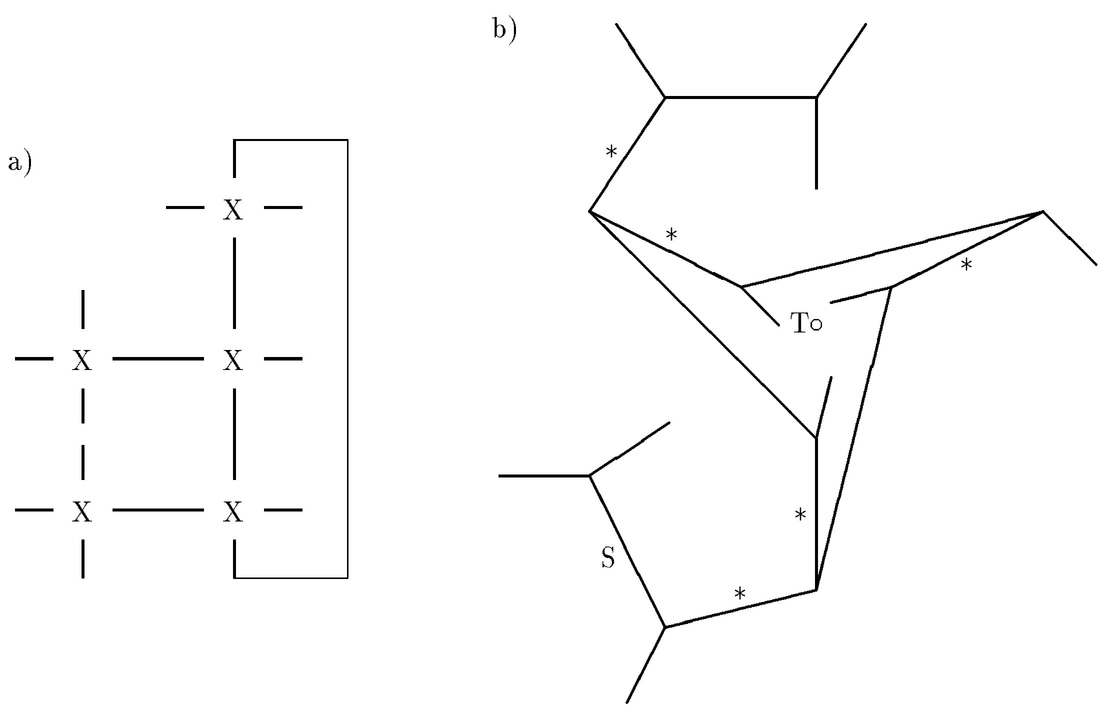
Lemma 3.1 *If a robot with a tactile sensor and target–ray location, but without compass, is a pushdown automaton, its behaviour in a homogeneous environment is periodic.*

This follows since the behaviour of a pushdown automaton working in scenes, in the infinite grid $\mathcal{G}_X$ can be simulated by a pushdown automaton working in certain 4–regular graphs whose half-edges are labelled by letters from the alphabet A (in a bijective manner, for every vertex). The concept of automaton working in half-edge–labelled graphs (compass labyrinths) is straigtforwardly defined, for details see [H]. An automaton in such a regular graph is working autonomously, since it always gets the same input information. And it is well–known that an autonomously working (deterministic) pushdown automaton always generates a finally periodic output sequence. □

Theorem 3.2 *If the behaviour of a robot with a tactile sensor and target–ray location, but without a compass, is periodic in a homogeneous environment, it does not solve the target–reaching problem.*

For such a robot, respectively for its sequence f, one can construct a *trap*. This is a scene consisting of X–fragments, essentially. They are linked and the boundaries are closed (this means their pieces are completed to simply closed paths) in such a way that the target is topologically reachable, but the robot walks along a cyclic path never reaching the target.

Figure 6:



The construction of a trap uses standard techniques of labyrinth theory. If the sequence f determines a path which enters only a certain finite part of the grid $\mathcal{G}_X$, the trap corresponds to that part. The boundaries are closed in the simplest way, each separately. The target can be located outside the convex hull of all boundary points.

Now let the sequence f determine a path which visits infinitely many vertices of the grid. Then we consider two vertices X_1 and X_2 in the grid, with a distance ≥ 3, which correspond to the same position in the period u of f. This means, starting from X_1, one reaches X_2 by walking a path corresponding to some $\hat{u}^k$ with $k \geq 1, \hat{u} = u_2 \cdot u_1$, where $u_1 \cdot u_2 = u$. Then we cut a suitable strip containing X_1 and X_2 out of the grid and paste its ends together in a loop around the target of the scene by identifying X_1 with X_2. Now the walk according to f becomes cyclic on this strip if the starting point is suitably chosen. The boundaries of the scene are simply closed like above.

Of course, a complete proof of the theorem would require a detailed treatment of the method of *cutting and pasting*. Also the *reduction modulo trees* should be mentioned and applied. Since these techniques have extensively been used in [BK] and [H], here we close with an example for a trap construction.

Let $f = \mathrm{d} \cdot (\mathrm{abdaa})^\omega$. Figur 6 a) shows the schema of the trap which consist of five X–fragments. The starting point lies in the lower left fragment. Figure 6 b) sketches a corresponding scene, where the presentation of X–fragments according to Figure 5 b) is used. The fragment, in which the starting point lies, is marked by S, the location of the target is fixed. Moreover, by stars we have marked the points at which two X–fragments are linked.

4 Exhaustive Search

For a scene $\mathcal{S} = (\mathcal{P}, T, S)$, let

$$G_{\mathcal{S}} = (V_{\mathcal{S}}, E_{\mathcal{S}})$$

be the following finite, directed graph.

1. The vertex set $V_{\mathcal{S}}$ consists of all vertices of the obstacles of $\mathcal{S}$, of the points T and S, and of all points v for which there exists a point P, with $P = S$ or P is a vertex of an obstacle of $\mathcal{S}$, such that the target ray $\vec{PT}$ at P does not point into the obstacle and v is the first point in which the target ray starting in P meets the boundary of some obstacle.

2. For vertices $v_1, v_2 \in V_{\mathcal{S}}$, there is a directed edge $(v_1, v_2) \in E_{\mathcal{S}}$ iff

   (a) v_1 and v_2 are neighbouring vertices of the same obstacle of $\mathcal{S}$ (i.e. v_2 can directly be reached from v_1 by walking along that boundary); or

   (b) $v_1 = S$ or $v_1 = P$, for some vertex P of an obstacle such that the ray $\vec{PT}$ at P does not point into the obstacle, and v_2 is the first point in which the target ray starting at v_1 meets the boundary of some obstacle or T; or

   (c) v_1 does not belong to the vertices of obstacles of the scene, but it is an endpoint of an edge defined by (b), and v_2 is one of the both endpoints of the side of the obstacle on which v_1 lies.

Figure 7 a) shows a scene $\mathcal{S}$, and Figure 7 b) sketches the graph $G_{\mathcal{S}}$ obtained from it. Here the dots and the thick lines correspond to vertices resp. edges of the scene (these edges can be used in both directions, in the graph), whereas the small circles and thin lines represent those vertices and directed edges which are caused by moves along the target ray. For a better representation, the latter vertices are shifted from their positions in the plane.

We remark that the vertices T and S always belong to the same component (of connectedness) of the graph $G_{\mathcal{S}}$ if T lies outside the obstacles of the scene.

The outdegree of any vertex in $G_{\mathcal{S}}$ is bounded by 3, and any path in this graph can be thought to be controlled by some word over the alphabet

$$C = \{b^+, b^-, t\}.$$

More precisely, for a vertex $v \in V_{\mathcal{S}}$, a word $w \in C^*$, and a letter $c \in C$, let

$$\mathrm{pos}(v, \Lambda) = v,$$

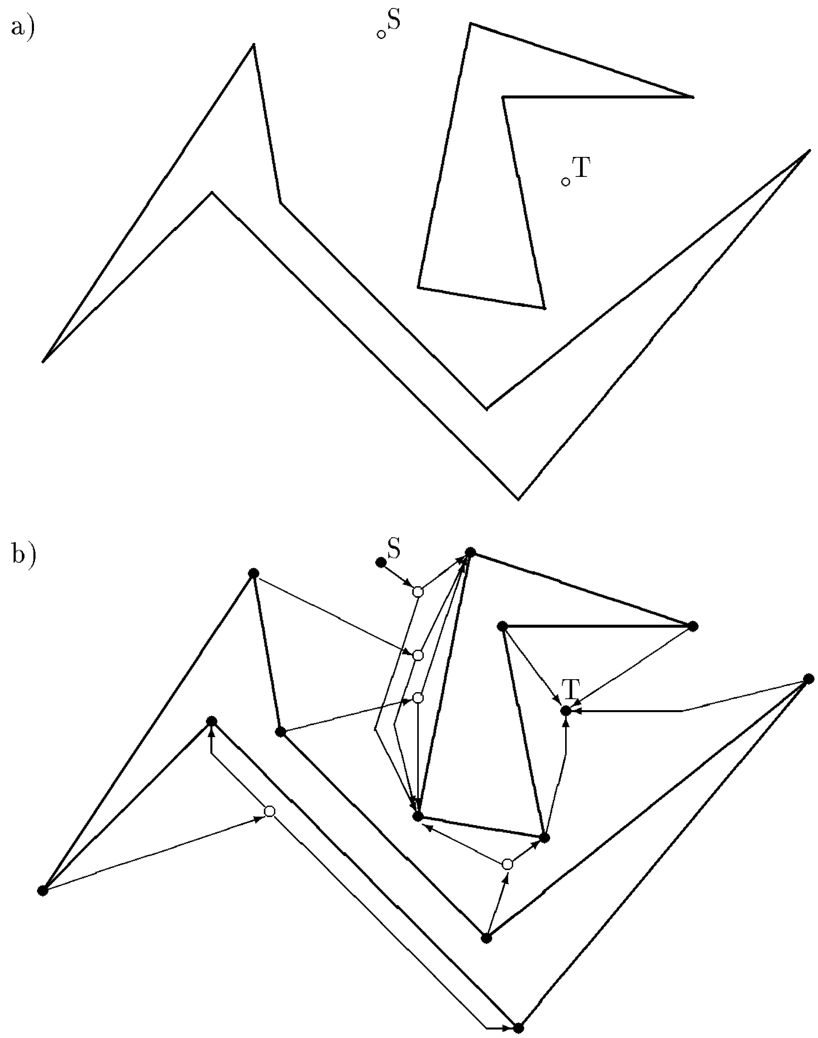
and

$$\mathrm{pos}(v, w \cdot c) = v'$$

iff

1. $c \in \{b^+, b^-\}$, $\mathrm{pos}(v, w)$ lies on the boundary of some obstacle, and v' is that vertex (of the scene) which is first reached from $\mathrm{pos}(v, w)$ by walking along that boundary according to the positive resp. negative orientation; or

Figure 7:



2. $c = t$, $\text{pos}(v, w)$ is a vertex of some obstacle of the scene, from which the target ray does not point into the obstacle, and v' is the first point in which the target ray starting from $\text{pos}(v, w)$ hits a boundary of an obstacle or T; or

3. $v' = \text{pos}(v, w)$ in all other cases (where the move instruction c cannot be applied in the sense of 1. or 2.).

Remark that the paths in $G_{\mathcal{S}}$, which are controlled by words from C^*, correspond to those walks in the scene $\mathcal{S}$, which can be performed by a robot with a tactile sensor and target–ray location if it never moves away from T on the straight line $\overleftrightarrow{ST}$.

Lemma 4.1 *Let $\mathcal{S}$ be a scene whose target does not belong to an obstacle. Then, to every vertex $v \in V_{\mathcal{S}}$, there is a control word $w_v \in C^*$ which leads from v to the target T, i.e. $pos(v, w_v) = T$.*

This is shown by induction on the number of obstacles of the scene $\mathcal{S}$. If there is no obstacle, the assertion holds trivially.

We assume, it holds for all scenes with $k \geq 0$ obstacles, and consider a scene $\mathcal{S}$ with $k+1$ obstacles. There is a vertex P_1 of some obstacle, say p_1, which has a minimal distance from the target T. Then $P_1 = T$, or the target is reachable from P_1 by a straight move. Therefore, for any vertex v_1 which belongs to p_1, there is a control word w_{v_1} leading from v_1 to the target, in the graph $G_{\mathcal{S}}$. This corresponds to a walk along p_1, possibly followed by a straight move to T.

Let v be an arbitrary vertex of $G_{\mathcal{S}}$. If it belongs to p_1, the assertion holds. Otherwise, we consider the scene $\mathcal{S}'$ obtained from $\mathcal{S}$ by removing the boundary p_1. Since $\mathcal{S}'$ has only k obstacles, there is a control word w'_v leading from v to the target T in the graph $G_{\mathcal{S}'}$. If the path controlled by w'_v in the graph $G_{\mathcal{S}}$ never meets vertices belonging to the boundary p_1, the assertion is proved. Otherwise, we obtain a control word w_v by concatenating a certain initial part of w'_v (up to reaching p_1) with a control word leading from the entered vertex on p_1 to the target. $\Box$

The length of the control word w_v can be bounded by the number of vertices of the graph $G_{\mathcal{S}}$, this is at most twice the number of vertices (of obstacles) in the scene $\mathcal{S}$.

Lemma 4.2 *If the target is topologically reachable in a scene $\mathcal{S}$, there is an $\mathcal{S}$–universal control word $w_{\mathcal{S}}$ such that, for every vertex $v \in V_{\mathcal{S}}$, the path starting at v and controlled by $w_{\mathcal{S}}$ meets the target T. More precisely, $pos(v, w') = T$ for some initial part w' of $w_{\mathcal{S}}$.*
The length of $w_{\mathcal{S}}$ can be bounded by the square of the number of vertices of the graph $G_{\mathcal{S}}$.

The universal control word can be constructed by induction using Lemma 4.1. Let the vertex set be enumerated,

$$V_{\mathcal{S}} = \{v_0, \cdots, v_l\}.$$

We define

$$w_0 = w_{v_0},$$

according to Lemma 4.1; this leads from v_0 to the target T.

Assume that w_i has been defined for some $i < l$ such that it leads through T, for every vertex from $\{v_0, \cdots, v_i\}$, taken as starting point. Let $v^\star_{i+1} = \mathrm{pos}(v_{i+1}, w_i)$, and

$$w_{i+1} = w_i \cdot w_{v^\star_{i+1}},$$

where the latter word leads from $v^\star_{i+1}$ to T, according to Lemma 4.1. Then w_{i+1} leads through T, for every starting vertex from $\{v_0, \cdots, v_i, v_{i+1}\}$. Therefore, the word

$$w_{\mathcal{S}} = w_l$$

has the properties claimed in the lemma. $\Box$

Theorem 4.1 *The target–reaching problem can be solved by a robot with target–ray location and tactile sensor, without using a compass, with the complexity $exp(O(n^2))$.*

By a *universal C–sequence*, we understand an infinite sequence over the alphabet C,

$$f_u \in \{b^+, b^-, t\}^\omega,$$

such that every word $w \in C^*$ is contained in f_u as a substring.

We assume that the computational power of the robot suffices to generate a universal C–sequence $f_u = (c_k)_{k\in\mathbf{N}}$. Using the tactile sensor and the target–ray location, in any scene $\mathcal{S}$, then it can walk according to this sequence, i.e. it takes the sequence of positions $(\mathrm{pos}(S, c_0c_1\cdots c_k))_{k\in\mathbf{N}}$, where S is the starting point. If it reaches the target, let the robot halt. By Lemma 4.2, this procedure solves the target–reaching problem.

A universal C–sequence f_u is obtained by consecutively producing all words over the alphabet C, according to the lexicographic ordering. By Lemma 4.2, for every n–vertex graph whose target T is topologically reachable, T has been reached by the walk according to that sequence f_u when all C–words of length n^2 have been occurred. This is the case after $\sum_{i=1}^{n^2} 3^i \leq n^2 \cdot 3^{n^2} = 3^{O(n^2)}$ breakpoints of move. $\Box$

5 Rotation Counting – the Simple Case

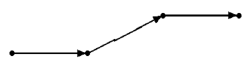
Now we are going to prepare the ingredients for a target–reaching algorithm with quadratic complexity. It is mainly based on rotation counting, a method which, for rectilinear mazes, was already used in [A] and has been developed in detail in [H].

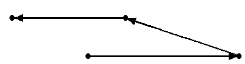
The *rotation index* $\mathrm{rin}(p)$ of a polygonal path $p = (P_1, \cdots, P_k)$ gives the angle by which the direction of move is changed in the course of running through p. More precisely,

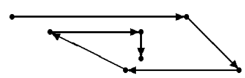
$$\mathrm{rin}(p) = \sum_{i=1}^{k-2} \alpha(\overrightarrow{P_i P_{i+1}}, \overrightarrow{P_{i+1} P_{i+2}}),$$

where $\alpha(r_1, r_2) \in [-\pi, \pi)$ is the angle from the ray r_1 to r_2, according to the counterclockwise orientation of the plane.

For example,

rin() $= 0$,

rin() $= \pi$,

rin() $= -\frac{9}{4}\pi$.

For a closed polygonal path $p = (P_1, \cdots, P_k, P_{k+1} = P_1)$, we also consider the *closed rotation index*

$$\overline{\mathrm{rin}}(p) = \mathrm{rin}(p) + \alpha(\overrightarrow{P_k P_1}, \overrightarrow{P_1 P_2}).$$

Theorem 5.1 *(Riemann,Hopf)*
For every simply closed polygonal path p, it holds

$$\overline{rin}(p) = \begin{cases} 2\pi & \textit{if the interior of } p \textit{ lies on the left–hand side of the path,} \\ -2\pi & \textit{if the interior of } p \textit{ lies on the right–hand side.} \end{cases}$$

Remember that, by Jordan's curve theorem, any simply closed path dissects the rest of the plane into two disjoint (open, simply connected) regions, the bounded *interior* and the unbounded *exterior* of p. Theorem 5.1 is a discrete version of the Riemann–Hopf theorem of turning tangents which is well–known from differential geometry. An elementary proof for the rectilinear case can be found in [H]. Here we will use it without any proof. □

A polygonal path p in $\mathcal{S}$ is said to be *boundary–following* if it is a subpath of some boundary p_i (with the same orientation). More precisely, if $p_i = (P_1, \cdots, P_k, P_{k+1} = P_1)$, then $p = (Q, P_{i+1}, \cdots, P_{i+l}, Q')$, with $1 \leq i \leq k, l \in \mathbf{N}, Q \in \overline{P_i P_{i+1}}$ and $Q' \in \overline{P_{i+l} P_{i+l+1}}$, where $P_{k+j} = P_j$ for $j > 1$, and Q' lies between Q and Q_{i+1} if $l = 0$.

Let P and Q be (different) points which belong to the same boundary of the given scene. By the *bow* from P to Q, denoted by $b(P, Q)$, we mean that boundary–following and loop–free path which starts in P and terminates with Q (when it is reached at the first time in walking along the boundary, starting in P).

Agreement. To simplify some definitions and explanations, in the sequel we suppose that the starting point and the target of the scene $\mathcal{S} = (\mathcal{P}, T, S)$ have the same y–coordinate, and the x–coordinate of S is less than that of T. This means, the straight line $\overleftrightarrow{ST}$ is horizontal, and S lies on the left of T. Moreover, we will suppose that no vertex of the obstacles of the family $\mathcal{P}$ belongs to the straight line $\overleftrightarrow{ST}$. So, whenever a boundary of an obstacle has a common point with $\overleftrightarrow{ST}$, this is a proper crossing point.

If a path $q = (Q_1, \cdots, Q_k)$ with $k \geq 3$, starts and ends on the straight line $\overleftrightarrow{ST}$ on the left of T, more precisely $Q_1, Q_k \in \overrightarrow{TS}$, its *target–directed rotation index* is defined by

$$\text{tdrin}(q) = \text{rin}(q) + \alpha(\overrightarrow{Q_1T}, \overrightarrow{Q_1Q_2}) + \alpha(\overrightarrow{Q_{k-1}Q_k}, \overrightarrow{Q_kT}).$$

In other words, if $Q, Q' \in \overleftrightarrow{ST}$, Q on the left of Q_1 and Q' on the right of Q_k, then $\text{tdrin}(q) = \text{rin}(Q, Q_1, \cdots, Q_k, Q')$. Since the rays $\overrightarrow{QQ_1}$ and $\overrightarrow{Q_kQ'}$ point in the same direction, we have $\text{tdrin}(q) = j \cdot 2\pi$, for some integer j.

In the remaining part of this section,we restrict ourselves to simple scenes. A scene $\mathcal{S} = (\mathcal{P}, T, S)$ is said to be *simple* (or an *s–scene*) if no boundary of the family $\mathcal{P}$ crosses the straight line $\overleftrightarrow{ST}$ on the right of T. Obviously, in simple scenes the target is always topologically reachable.

By an *s–swing* in some s–scene, we understand a polygonal path $q = (Q_1, \cdots, Q_k)$, with $k \geq 3$, such that

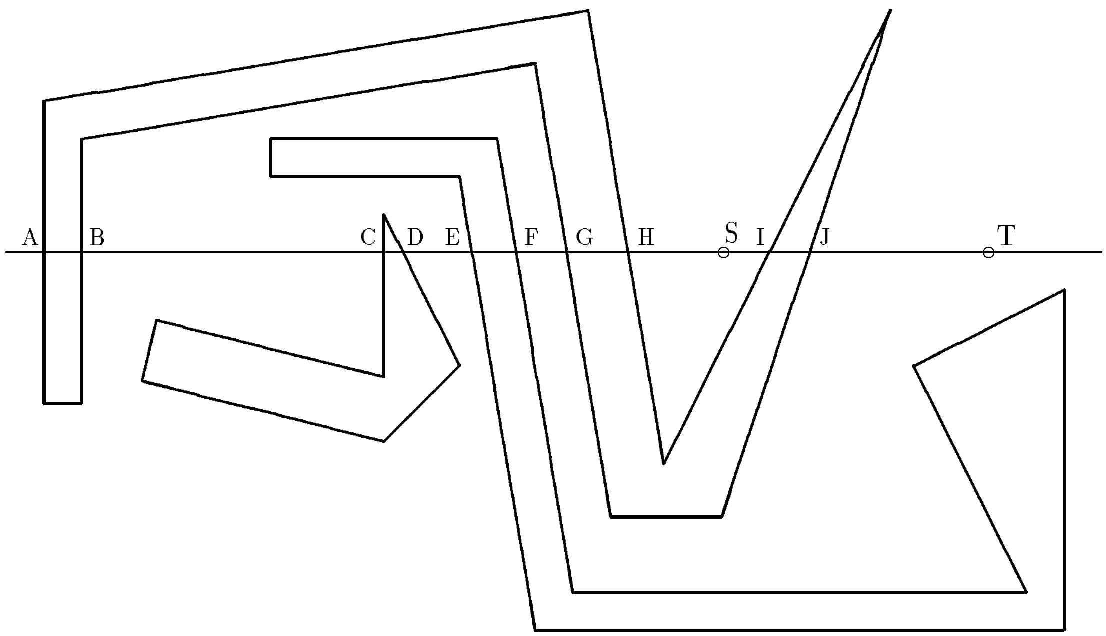
- q is boundary–following;
- it starts and ends on $\overleftrightarrow{ST}$,
  i.e. $Q_1, Q_k \in \overleftrightarrow{ST}$;
- q starts with a downward move and ends with an upward one,
  i.e. the y–coordinates of Q_2 and Q_{k-1} are less than those of S and T ; and
- $\text{tdrin}(q) = 0$.

An s–swing is said to be *regular* if there is no proper initial part which is an s–swing, too.

Figure 8 illustrates the concepts introduced so far. Examples for s–swings are the bows $b(A, B), b(A, J), b(A, H), b(C, D), b(E, F), b(G, J), b(G, H), b(I, B), b(I, J)$ in this simple scene. The s–swings $b(A, B), b(C, D), b(E, F), b(G, J)$ are regular. There are only finitely many regular s–swings, in any scene. By Theorem 5.1, it even follows that there are only finitely many s–swings.

Lemma 5.1 *Let $\mathcal{S}$ be a simple scene. If $P \in \overleftrightarrow{ST} \cap p_i$, for some boundary p_i which crosses $\overleftrightarrow{ST}$ downwards at P (equivalently, the target ray points into the i–th obstacle at P), then there is just one regular s–swing starting at P, and that is loop–free.*

Figure 8:



Let the suppositions of the lemma be fulfilled. Since the ray $\overrightarrow{PT}$ points into the i–th obstacle at P, but T does not belong to that obstacle, there is a first point Q on $\overrightarrow{PT}$ in which the ray crosses the boundary p_i. The set $\mathrm{bow}(P,Q) \cup \overline{QP}$ is (the carrier of) a simply closed polygonal path, and by Theorem 5.1 it follows that $\mathrm{tdrin}(b(P,Q)) = 0$. Hence, the bow $b(P,Q)$ is a loop–free s–swing. It has just one initial part being a regular swing, and that is loop–free, too. $\Box$

For polygonal paths $q = (Q_1, \cdots, Q_k)$ and $q' = (Q'_1, \cdots, Q'_{k'})$, we say that q is *concatenable* with q' if Q'_1 lies on the right of Q_k and the interior of the straight line segment $\overline{Q_k Q'_1}$ does not contain a point of an obstacle. The latter means that

$$\overline{Q_k Q'_1} \cap (\bigcup_{i=1}^{m} p_i) \subseteq \{Q_k, Q'_1\}.$$

In this case, the polygonal path

$$q \oplus q' = (Q_1, \cdots, Q_k, Q'_1, \cdots, Q'_{k'})$$

is called the *concatenation* of q with q'. If q and q' are s–swings, we have $\mathrm{tdrin}(q \oplus q') = 0$. The concatenation of finitely many paths is straightforwardly defined.

By a [*regular*] *s–wave*, we mean a non–empty finite concatenation of [regular] s–swings.

The following theorem gives the basis of our target–reaching algorithm.

Theorem 5.2 *A regular s–wave in a simple scene cannot be concatenable with itself.*

Figure 8 shows that the supposition of the regularity in the theorem is essential. The s–wave $b(I,B) \oplus b(C,D) \oplus b(E,F) \oplus b(G,H)$ is concatenable with itself.

The proof of the theorem follows from the fact that, roughly speaking, in a simple scene any two regular s–swings cannot cross each other. More precisely, let $q = b(Q_1, Q_2)$ and $q' = b(Q'_1, Q'_2)$ be regular s–swings in a simple scene. If they are different, but have a common point, one of the both, say q', must start in the interior of the other one, i.e. $Q'_1 \in b(Q_1, Q_2) \setminus \{Q_1, Q_2\}$ [2]. But then q' must terminate at a point within q inclusively Q_2, i.e. $Q'_2 \in b(Q_1, Q_2)$.

To prove the latter claim, remark first that

$$0 = \text{tdrin}(b(Q_1, Q_2)) = \text{tdrin}(b(Q_1, Q'_1)) + \text{tdrin}(b(Q'_1, Q_2)).$$

Moreover, $\text{tdrin}(b(Q_1, Q'_1)) = j \cdot 2\pi$, with an integer j. Now we consider the sequence of the target–directed rotation indices of all initial parts of $b(Q_1, Q_2)$ which terminate with an upward move on $\overleftrightarrow{ST}$ (ordered by increasing lengths of the bows). The difference between two consecutive values of this sequence always belongs to $\{-2\pi, 0, 2\pi\}$. Since $Q'_1 \neq Q_2$, it follows $j < 0$ and $\text{tdrin}(b(Q'_1, Q_2) = (-j) \cdot 2\pi$. Q'_2 is the first point $Q \in \overleftrightarrow{ST}$ with $\text{tdrin}(b(Q'_1, Q)) = 0$, along the boundary–following path starting at Q'_1. Therefore, Q_2 cannot lie before Q'_2, along a boundary–following path starting at Q_1.

Assume, there would be a regular s–wave, say $q = (Q_1, \cdots, Q_k)$, which is concatenable with itself. Different s–swings of q could have common points, but they cannot cross each other. Therefore, by shifting (by sufficiently small amounts in the plane) the points in q when they occur repeatedly, from $\hat{q} = (Q_1, \cdots, Q_k, Q_1)$ we obtain a simply closed polygonal path q' with the same closed rotation index,

$$\overline{\text{rin}}(q') = \overline{\text{rin}}(\hat{q}) = \text{tdrin}(q) = 0.$$

This is a contradiction to Theorem 5.1. □

Theorem 5.3 *The target–reaching problem for simple scenes is solvable by a robot with a tactile sensor, target–ray location and compass, with the complexity $O(n^2)$.*

The algorithm is rather simple. Starting at some point S, let the robot perform the following progam. Here we simply write "regular swing" instead of "regular s–swing", because the concept of swing will be introduced for arbitrary scenes in the next section, in such a way that an s–swing is a swing in an s–scene.

```
Program          { target-reaching by regular swinging } ;
        repeat   from the current position P follow the target-ray
                     up to reaching T or a boundary of an obstacle ;
                 if T is reached, then halt (``target reached'') ;
                 if a point Q of some boundary is reached, then move
                     to the end of the regular swing starting at Q ;
        until    T is reached.
```

[2] We hope the reader will not be confused if we sometimes characterize a path as a sequence of points (its vertices), and sometimes identify a path with its carrier set.

It is obvious that a robot of the given type is able to perform this program. After reaching a point on a boundary of an obstacle, the program controls a walk along a regular s–wave up to the target T. Since there are only finitely many regular s–swings but there is no cyclic regular s–wave by Theorem 5.2, the program always terminates.

The number of regular s–swings in the scene is bounded by the number of crossing points of the boundaries of obstacles with the straight line $\overleftrightarrow{ST}$. That is bounded by the number n of vertices of the scene. The number of vertices of any regular s–swing is also bounded by n, because it is loop–free. Therefore, our algorithm has a complexity of $O(n^2)$. $\square$

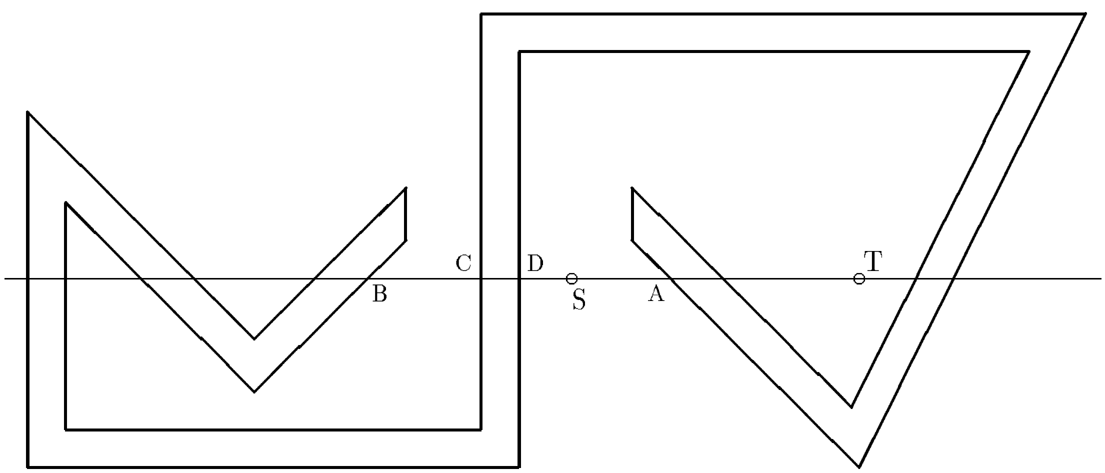
6 Rotation Counting – the General Case

Let the agreement from the previous section, that $\overleftrightarrow{ST}$ is horizontal and does not contain vertices of obstacles, also hold for the not necessarily simple scenes we are going to consider now.

Originally, the concept of swing for arbitrary scenes has been defined like that of s–swing, with the restriction that it should both start and terminate on the left–hand side of the target T, on the straight line $\overleftrightarrow{ST}$. Such a condition seems to be necessary in order to secure that, from an endpoint of a regular swing, by following the target ray, one reaches either the target or a starting point of another regular swing.

Unfortunately, with this definition of [regular] swing, the assertion of Theorem 5.2 would not hold. As it has been pointed out by R. Klein [personal communication], there would be cyclic regular waves in this case. Figure 9 shows an example. The polygonal path $b(A, B) \oplus b(C, D)$ would be concatenable with itself. To avoid this dilemma, in running through the scenes, we observe the balance of the crossings of $\overleftrightarrow{ST}$ on the right of T.

Figure 9:



As usual, $\mathcal{S} = (\mathcal{P}, T, S)$ is the given scene. By $\overrightarrow{ST}^{r}$, we denote the open ray of all points of $\overleftrightarrow{ST}$ which lie on the right of T (T is excluded).

For a polygonal path q (which starts and terminates on $\overleftrightarrow{ST}$, but has no further vertex on $\overleftrightarrow{ST}$, let $\mathrm{cru}(q)$ denote the number of upward crossings with $\overrightarrow{ST}^{r}$. This is the number of occurrences of points $P \in q \cap \overrightarrow{ST}^{r}$ in q such that the y–coordinates are increasing in running through P along q. Analogously, let $\mathrm{crd}(q)$ denote the number of downward crossings with $\overrightarrow{ST}^{r}$ in running through q. If q starts and/or terminates on $\overrightarrow{ST}^{r}$, the starting point and/or the endpoint have to be counted as an upward resp. downward crossing, too.

The *crossing balance* of the path q is defined by

$$\mathrm{crbal}(q) = \mathrm{cru}(q) - \mathrm{crd}(q).$$

By a *swing* in a scene $\mathcal{S}$, we understand a polygonal path $q = (Q_1, \cdots, Q_k), k \geq 3$, such that

- q is boundary–following;
- q starts with a point from $\overleftrightarrow{ST} \setminus (\overrightarrow{ST}^r \cup \{T\})$ and terminates with a point from $\overleftrightarrow{ST} \setminus \overrightarrow{ST}^r$ (T is excluded as starting point but could be the endpoint of a swing);
- q starts with a downward move and terminates with an upward move;
- $\mathrm{tdrin}(q) = 0$; and
- $\mathrm{crbal}(q) = 0$.

A swing is said to be *regular* if there is no proper initial part which is a swing, too.

Lemma 6.1 *Let $\mathcal{S}$ be a scene whose target does not belong to an obstacle. If P is a point from $(\overleftrightarrow{ST} \setminus (\overrightarrow{ST}^r \cup \{T\})) \cap\ p_i$, for some boundary p_i which crosses $\overleftrightarrow{ST}$ downwards at P, then there is just one regular swing starting at P, and that is loop–free.*

The proof is analoguos to that of Lemma 5.1. We consider the first point $Q \neq P$, in which the ray $\overrightarrow{PT}$ crosses p_i. Such a point does exist in $\overleftrightarrow{ST} \setminus \overrightarrow{ST}^r$, since T lies in the exterior of p_i. Then $b(P,Q) \cup \overline{QP}$ determines a simply closed path q with the interior on the left–hand side. By Theorem 5.1, it follows $\mathrm{tdrin}(b(P,Q)) = 0$.

Now we consider the sequence of the crossing points of q with the ray $\overrightarrow{ST}^r$, ordered according to their occurrences along that ray. The first of these crossing points (if there is one) is downward, since T lies in the exterior of q. Then the next one must exist and is upwards directed, and so on. So we obtain a sequence of pairs of downward and upward crossings, and it follows $\mathrm{crbal}(b(P,Q)) = 0$.

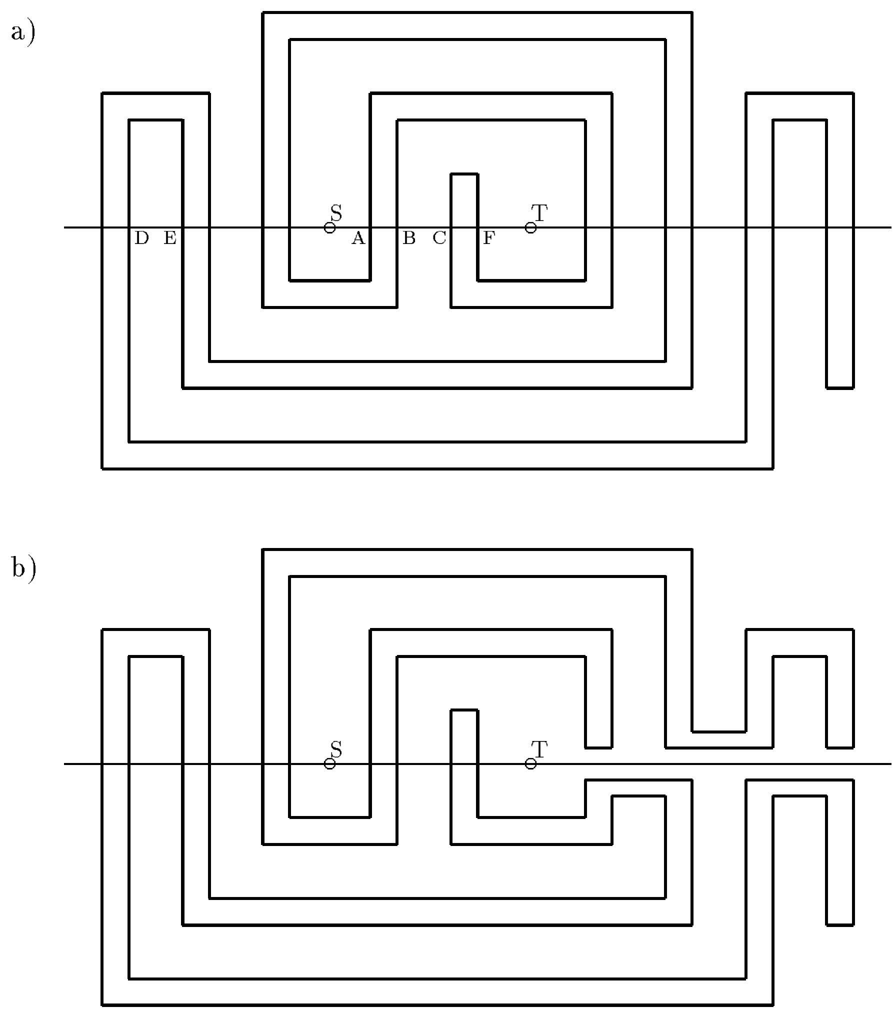
Therefore, $b(P,Q)$ is a loop–free swing, and it has just one initial part being a regular swing. □

By a [*regular*] *wave*, we mean a non–empty concatenation of finitely many [regular] swings.

Theorem 6.1 *In a scene whose target does not belong to an obstacle, a regular wave cannot be concatenable with itself.*

Unfortunately, the proof of Theorem 5.2 cannot be modified to show Theorem 6.1. There are scenes with pairs of regular swings which cross each other. More precisely, such pairs cause crossings of regular waves containing them. Figure 10 a) gives an example. The regular wave $b(A,B) \oplus b(C,D)$ crosses itself in the common part $b(A,D)$ of the both swings. Nevertheless, the regular wave $b(A,B) \oplus b(C,D) \oplus b(E,F)$, prolongated by $\overline{SA}$ and $\overline{FT}$, leads from S to T.

Figure 10:



Our proof of Theorem 6.1 works as follows. We describe a construction modifying an arbitrary scene $\mathcal{S}$, with a topologically reachable target, into a simple scene $\mathcal{S}'$ such that a regular wave concatenable with itself in $\mathcal{S}$ would yield a regular wave in $\mathcal{S}'$, i.e. a regular s–wave, concatenable with itself. But such one cannot exist, by Theorem 5.2.

The transformation of $\mathcal{S}$ into $\mathcal{S}'$ consists in cutting the boundaries, in all points where they cross the ray $\overrightarrow{ST}^{r}$, and pasting the free pieces together in a suitable way. The links between two upper resp. lower free pieces are described by $\cup$–pairs resp. $\cap$–pairs of crossing points.

(A, B) is called a $\cup$–*pair* if $A, B \in \overrightarrow{ST}^{r} \cap p_i$, for some boundary p_i of the given scene $\mathcal{S}$, such that p_i crosses $\overrightarrow{ST}^{r}$ downward in A but upward in B, and $b(A, B)$ is the shortest boundary–following path q starting in A such that $\mathrm{crbal}(q) = 0$. Remember that A and B

have to be considered as crossing points too, hence $A \neq B$.

Analogously, (A, B) is a $\cap$*–pair* if $A, B \in \overrightarrow{ST}^r \cap p_i$, for some boundary p_i which crosses $\overrightarrow{ST}^r$ upwards in A and downwards in B, and $b(A, B)$ is the shortest bow starting at A with the crossing balance equal to 0.

Lemma 6.2 *If the target T does not belong to an obstacle, every crossing point $P \in \overrightarrow{ST}^r \cap \bigcup_{i=1}^m p_i$ belongs to just one $\cup$–pair and just one $\cap$–pair.*

From the definition it follows immediately that any point can belong to at most one $\cup$–pair and at most one $\cap$–pair. Let P be a point in which p_i crosses the ray $\overrightarrow{ST}^r$ downwards. Since the interior of p_i lies on the left of the path and is bounded, there is a point on the right of P in which p_i crosses $\overrightarrow{ST}^r$ upwards. If Q denotes the first such point along $\overrightarrow{ST}^r$, $b(P, Q) \cup \overline{QP}$ determines a simply closed path q, and like in the proof of Lemma 6.1 we see that $\mathrm{crbal}(b(P, Q)) = 0$. Therefore, there is a point Q' such that (P, Q') is a $\cup$–pair.

But $b(Q, P) \cup \overline{PQ}$ determines a simply closed path too, and it holds $\mathrm{crbal}(b(Q, P)) = 0$. Thus there is a $\cap$–pair (Q'', P).

The proof for an upward crossing point P is analogous. □ (lemma)

Lemma 6.3 *Let (A_1, B_1) and (A_2, B_2) be two $\cup$–pairs or $\cap$–pairs, but both of the same type. Then A_2 lies between A_1 and B_1 if and only if B_2 does.*

This nesting property of $\cup$–pairs resp. $\cap$–pairs is fundamental for the pasting procedure in the proof of the theorem.

The proof of the lemma is by induction on the number of further crossing points along the bow $b(A_1, B_1)$. We consider the $\cup$–case only.

If there are no such further crossing points, the assertion of the lemma holds, for all $\cup$–pairs (A_2, B_2). Indeed, in this case $b(A_1, B_1) \cup \overline{B_1 A_1}$ determines a simply closed polygonal path. By Jordan's curve theorem, if $A_2 \in \overline{A_1 B_1} \setminus \{A_1, B_1\}$, then $B_2 \in \overline{A_1 B_1} \setminus \{A_1, B_1\}$, and conversely. Moreover, also the bows $b(A_2, B_2)$ do not contain further crossing points.

Suppose that the assertion holds if there are at most k crossing points along the bow $b(A_1, B_1)$. Assume that we have a $\cup$–pair (A_1, B_1) with $k+1 \geq 0$ further crossing points on the bow now. We consider the first point P along $b(A_1, B_1)$ which belongs to a downward crossing such that the next crossing point Q marks an upward crossing. Then (P, Q) is a $\cup$–pair without further crossing points on $b(P, Q)$. Therefore, all $\cup$–pairs with elements on $\overline{PQ}$ are completely contained in $\overline{PQ}$ and either nested or independent, mutually.

Then the bow $b(P, Q)$ and all bows $b(P', Q')$ belonging to $\cup$–pairs (P', Q') lying on $\overline{PQ}$ can be "lifted". More precisely, these bows together with sufficiently small surrounding pieces of the corresponding boundaries are replaced by straight line segments above $\overleftrightarrow{ST}$ such that no crossings are caused. This modification of the scene does not change the arrangement of the other crossing points. So the number of further crossing points along $b(A_1, B_1)$ has been decreased, and the supposition of induction applies. □ (lemma)

Lemma 6.4 *Let (A, B) be a $\cup$–pair or a $\cap$–pair, and q be the simply closed polygonal path* *determined by* $b(A,B) \cup \overline{BA}$ *(with the orientation of the bow). Then it holds*

$$\overline{rin}(q) = \begin{cases} 2\pi & \textit{if } (A,B) \textit{ is a } \cup\textit{-pair and } A \textit{ lies on the left of } B, \\ & \textit{or } (A,B) \textit{ is a } \cap\textit{-pair and } A \textit{ lies on the right of } B, \\ -2\pi & \textit{if } (A,B) \textit{ is a } \cup\textit{-pair and } A \textit{ lies on the right of } B, \\ & \textit{or } (A,B) \textit{ is a } \cap\textit{-pair and } A \textit{ lies on the left of } B. \end{cases}$$

This can be proved analogously to Lemma 6.3, by induction on the number of further crossing points along $b(A,B)$. The initial step of induction, where $b(A,B)$ contains no further crossing points, follows from Theorem 5.1 (Riemann–Hopf). The induction step is easier than for Lemma 6.3, because we only have to consider the bow $b(A,B)$ and no other boundary pieces of the scene. We omit the details. □ (lemma)

Now we can prove the theorem according to the idea sketched already. Let be given a scene $\mathcal{S}$ whose target does not belong to an obstacle. The scene $\mathcal{S}'$ is obtained from $\mathcal{S}$ by removing sufficiently small segments of the boundaries around the crossing points with the ray $\overrightarrow{ST}^{r}$. The free pieces are pastet together by straight line segments corresponding to the $\cup$–pairs and $\cap$–pairs.

More precisely, for every $\cup$–pair (A,B), the free pieces of the boundary at points A and B are connected by a straight line segment lying above (and parallel to) the straight line $\overleftrightarrow{ST}$. If another $\cup$–pair (A',B') is nested in (A,B), its connecting straight line segment must lie above that of (A,B). So it is secured that no crossing points between the parts of boundaries are caused. Analogously, the free ends corresponding to a $\cap$–pair (A,B) are connected by a straight line segment lying (in a sufficiently small distance) below $\overleftrightarrow{ST}$, such that every segment corresponding to another $\cap$–pair which is nested in (A,B) is lower located. We also omit the formal details of this construction. For an illustration, the reader is referred to Figure 10, where a) shows a scene $\mathcal{S}$ and b) the modified scene $\mathcal{S}'$.

In this way, from $\mathcal{S}$ we obtain a simple scene $\mathcal{S}'$. Assume that there would be a regular wave

$$w = s_1 \oplus \cdots \oplus s_l, \quad l \geq 1,$$

concatenable with itself in the scene $\mathcal{S}$. In the new scene $\mathcal{S}'$, from the regular swings $s_i, 1 \leq i \leq l$, we obtain regular swings s_i', each with the same starting and terminating point as its preimage s_i. Indeed, if the starting point A of a bow $b(A,B)$, for some $\cup$–pair or $\cap$–pair (A,B), belongs to a regular swing s_i, then also the whole bow must be contained in that swing. The same holds for the endpoint B of such a bow. Thus the replacements of the bows by straight line segments do not influence the starting or terminating points of the swings. Moreover, by Lemma 6.4, the rotation indices of the swings (and of their initial parts terminating on $\overleftrightarrow{ST} \setminus \overrightarrow{ST}^{r}$) are not changed.

As a regular wave in a simple scene,

$$w' = s_1' \oplus \cdots \oplus s_l'$$

would be a regular s–wave, and concatenable with itself. This is a contradiction to Theorem 5.2 . □ (theorem)

Theorem 6.2 *The target–reaching problem (for arbitrary scenes whose targets do not belong to obstacles) is solvable by a robot with a tactile sensor, target–ray location and compass, with the complexity* $O(n^2)$.

This can be shown using the program given in the proof of Theorem 5.3. The correctness and complexity estimation follow like in that proof, using Lemma 6.1 and Theorem 6.1. □

7 Discussion

We have studied the target–reaching abilities of point robots equipped with tactile sensors, with facilities for target–ray localization and, possibly, with compasses.

If no compass is available, the behaviour of a target–reaching procedure cannot be periodic in homogeneous environments. On the other hand, the exhaustive search, as an extremely non–periodic procedure, solves that problem by trying blindly all possible ways in an arbitrary scene. These results concern the program complexity of robots. With respect to our step–number complexity, we only have the upper bound $\exp(O(n^2))$ by exhaustive search, but no non–trivial lower bound. Can the target–reaching problem be solved with a subexponential complexity, without using a compass?

Also for a robot with compass, some questions remain open. Does there exist a target–reaching algorithm with a complexity below $O(n^2)$, perhaps for interesting special classes of scenes? What about the simple scenes in the sense of Section 5, for example? We remark that, for scenes with only one obstacle, there is a linear target–reaching algorithm, even without using a compass.

In [BK] also the *wall problem* has been considered. Here the target is a straight line, let's say in north–south direction, on the right of the starting point and the obstacles. In this case, the target ray perceptible by the robot is perpendicular to the target wall. It is not hard to modify the exhaustive search correspondingly. Moreover, now the rotation–counting method correponds straightforwardly to the regular–swinging method used to escape from (rectilinear) two–dimensional mazes, see [A], [H]. So, for the wall problem, we obtain results which are analogous to those for the target–point task.

More important is the problem to reach a target point in an *included scene* $(\mathcal{P}, T, S, \overline{p})$. Here let $(\mathcal{P}, T, S)$ be a scene in the usual sense, and $\overline{p}$ is a simply closed polygonal path including this scene. This means, the interior of $\overline{p}$ lies on the right–hand side of the path and contains the points S and T and all obstacles. Now the rotation–counting method cannot immediately be applied, since for a downward crossing point $P \in \overline{p} \cap \overleftrightarrow{ST}$, on the left of T, there is not necessarily a (regular) swing starting in P.

On the other hand, an included scene naturally determines a finite, connected, plane graph such that the robot can implement graph–searching algorithms. And for finite, connected, (rectlinearly) embedded graphs there is a searching algorithm based on rotation counting, see [H]. Unfortunately, it is not immediately applicable to our problem since it uses a pebble, i.e. a marker for vertices of the graph. But there is another consolation: For the practically essential case, where the interior of $\overline{p}$ forms a convex set in the plane, the program from Section 5 works correctly. More generally, the program is successful if $\overline{p}$ crosses the straight line $\overleftrightarrow{ST}$ in two points only. Indeed, in this case a regular wave starting from S never hits the boundary $\overline{p}$.

The concept of step–number complexity used in this paper allows a rather simple evaluation of the algorithms presented here. A more detailed view could consider the lengths of paths. Moreover, the competitive ratio, as mentioned in Section 2, should be investigated, even if it cannot be bounded by a constant. In [BRS], a corresponding "figure of merit" for a robot has been proposed.

Acknowledgement

I am grateful to Rolf Klein both for the encouragement to try the bridge from labyrinth problems to on–line motion planning, and for the stimulating discussions from which some basic ideas of this paper arose.

References

[A] G. Asser. Bemerkungen zum Labyrinth–Problem. EIK 15 (3), 1977, 203–216.

[B] L. Budach. Automata and labyrinths. Math. Nachrichten 86, 1978, 195–282.

[BBFY] E. Bar-Eli, P. Berman, A. Fiat, P. Yan. On–line navigation in a room. 3rd SODA, 1992, 237–249.

[BK] M. Blum, D. Kozen. On the power of compass. 19th FOCS, 1978, 132–142.

[BRS] A. Blum, P. Raghavan, B. Schieber. Navigation in unfamiliar geometric terrain. 23rd STOC, 1991, 494–504.

[DKP] X. Deng, T. Kameda, C. Papadimitriou. How to learn an unknown environment. 32nd FOCS, 1991, 298–303.

[H] A. Hemmerling. Labyrinth problems – labyrinth-searching abilities of automata. Teubner–Texte zur Mathematik v. 114, Leipzig 1989.

[HA] Y.K. Hwang, N. Ahuja. Gross motion planning – a survey. ACM Computing Surveys v. 24, 3, 1992, 219–291.

[K] R. Klein. Walking an unknown street with bounded detour. 32nd FOCS, 1991, 304–313.

[LS] V.J. Lumelsky, A.A. Stepanov. Path–planning strategies for a mobile automaton moving amidst unknown obstacles of arbitrary shape. Algorithmica 1987, 2, 403–430.

[PY] C.H. Papadimitriou, M. Yannakakis. Shortest path without a map. TCS 84, 1991, 127–150. (see also 16th ICALP, 1989, 610–620)

[Y] C.-K. Yap. Algorithmic motion planning. In: Algorithmic and geometric aspects of robotics. v. 1, Hillsdale and London 1987, 95–143.