Planning Paths of Minimal Curvature

Jürgen Sellen *
Courant Institute of Mathematical Sciences
New York University

Abstract

We consider the problem of planning curvature constrained paths amidst polygonal obstacles, connecting given start and target configurations. Let the critical curvature R_c be the minimal curvature for which a constrained path exists. We describe an algorithm, which approximates the critical curvature and finds a corresponding path. Further, we give an efficient decision procedure to determine if there exists a path satisfying a given curvature constraint R, with running time polynomial in $|R - R_c|/R$.

1 Introduction

Traditional research in path planning focuses on the planning of collision-free paths for "free-flying" objects. In recent years, research has begun to address the subject of planning trajectories under non-holonomic constraints, such as constraints on velocity or curvature. Although adding more complexity to the problem, such constraints are necessary to appropriately model planning problems in real world applications.

One of the most basic problems in this area is the problem of planning paths for a point-sized mobile robot with minimum turning radius, moving among polygonal obstacles in the plane. This problem is regarded as a first step towards modeling the actual kinematics of a car. It translates to planning smooth paths of bounded curvature from a given initial to a given final configuration.

1.1 Problem Statement

In the bounded curvature problem, configurations correspond to tuples specifying position and orientation of the moving point. Orientations are given by unit vectors $u \in \mathbf{R}^2$.

The *basic planning problem* can be formulated as follows: given initial and final configurations $S = (P_S, u_S)$, $T = (P_T, u_T)$, a polygonal environment $E \subseteq \mathbf{R}^2$, and a curvature bound $R > 0$, find a C^1-path $w : [0, l] \rightarrow \mathbf{R}^2$ satisfying

*Supported by a postdoctoral fellowship from the DAAD, Germany.

(1) $w(t) \in E \;\; \forall\, t \in [0, l]$,

(2) $||\dot{w}(t)|| = 1$,
$||\dot{w}(t_2) - \dot{w}(t_1)|| \leq R^{-1}|t_2 - t_1| \;\; \forall\, t, t_1, t_2 \in [0, l]$,

(3) $(w(0), \dot{w}(0)) = S, (w(l), \dot{w}(l)) = T$.

The environment E is given by disjoint polygonal obstacles, with a total of n vertices. For simplicity, we shall assume that S and T coincide with obstacle vertices. The parameter R is called the *minimum turning radius* of the robot or the *maximal curvature* of the unit speed path w. We denote a path w satisfying all above constraints as *R-constrained.*

The *shortest path problem* in this context is to find an R-constrained path with minimal Euclidian length l. Another criterium to optimize smooth paths is the curvature itself. This is motivated by the fact that the speed at which we can follow the planned path will generally depend on the maximal curvature (e.g.: pipe or track layout). Given again S, T and E, we define the *critical curvature* R_c to be the supremum of $R \in [0, \infty]$, for which an R-constrained path w exists.

In contrast to the shortest path problem, there are situations in which no finite length path satisfying the constraint R_c exists. However, in every ε-neighborhood of R_c we can find a feasible finite length path. Thus approximation problems concerning R_c are still well-posed.

See figure 1: The shown path from S to S has critical curvature R_c. For any radius $R < R_c$, T can be reached from S by a circling path, following arcs touching $e1$, $v2$, $e2$, $v1$, $e1$, and so on. But at R_c, only points on $e1$ below S can be reached from S.

In this paper, we present an algorithm to approximate R_c, and to construct an R-constrained path with $|R - R_c| \leq \varepsilon R_c$, for given $\varepsilon > 0$. Further, we shall develop an efficient algorithm to plan an R-constrained path in running time dependent on the *relative width* $W = |R - R_c|/R$. This reflects the observation, that the basic planning problem is intuitively easy if W is large, but hard if W is small.

The methods used to obtain these algorithms are fairly general and should apply to similar problems, e.g. in kinodynamic motion planning. For simplicity in this abstract, we do not focus on optimizing paths according to length. However, the presented technique can be extended to guarantee that the solutions found are shortest in an approximate sense.

1.2 Previous Work

The mathematical foundation for the algorithmic treatment of bounded curvature problems has been laid in the 60's, by a paper of Dubins [Du] characterizing the shortest R-constrained trajectories between start and target configurations in the unrestricted plane. The main result consists in the reduction of the set of possible solutions to a discrete set of *canonical trajectories.* Recent research [RS,BCL] extends Dubin's work by allowing cusps (i.e. backing up), and gives alternative proofs.

The basic planning problem in (polygonal) environments has first been examined in [La1], giving a non-complete approach. Gordon and Wilfong [FW] describe an exact decision procedure, with a running time exponential in both the number of vertices and the bitsize of coordinates.

Jacobs and Canny [JC1,JC2] describe an approximation algorithm to compute shortest R-constrained paths. The grid-based algorithm finds an approximate shortest path if this path is sufficiently *robust* to allow for small changes in orientation (resp., position) along the path without causing collisions. The worst case running time is $O(\frac{n^3}{\delta}\log n + (\frac{n}{\delta})^2 \log(\frac{n}{\delta}))$, where δ is a measure for both the closeness of the approximation and the robustness of the path. The paper also presents a quadtree-based approach to optimize paths with regard to robustness.

A survey of this approach, as well as additional material on curvature constrained motion planning, can be found in [La2].

1.3 Outline and Results

The key idea in our approach is to construct an algorithm $A(R, \delta)$, which finds a feasible $(R - \delta)$-constrained path if there exists a feasible R-constrained path.

The algorithm follows in principle Papadimitriou's grid-based approach to the 3-dimensional Euclidian shortest path problem [Pa], revisited in [CSY]:

Applying Dubins' results, we can confine ourselves to *canonical paths* that connect *contact configurations* by canonical trajectories. The contact configurations arise as a contact of the moving point at an obstacle wall or at an obstacle corner. The feasible configurations in either situation can be described by a single parameter.

We subdivide each contact parameter uniformly into segments. To stress analogy to [Pa], we call two segments *visible* if they contain configurations which can be connected by a collision-free canonical trajectory. The main part of the algorithm is to compute the visibility relationship between all segments. As in [CSY], this can be done efficiently by an interesting kind of plane sweep. Finally, we interpret the segments as nodes of a visibility graph G, and search for a (shortest) path $\tilde{w}$ in G connecting the segments corresponding to S and T.

Section 2 gives a detailed description of this approach.

If we replace the edges of $\tilde{w}$ by canonical trajectories that realize the visibility, we get a path w with kinks, violating the smoothness condition. The remaining part is to show, that we can smoothen the path w by using a slightly higher curvature in the neighborhood of the kinks. This finally gives a feasible $(R - \Delta)$-constrained path.

The necessary grid-size to obtain $\Delta = \delta$ is derived in section 3. It depends, as well as the approximation error in [JC1,JC2], on the smallest distance l_{min} between obstacle features (for an exact definition see section 3). In fact, our approach relates the robustness notion in [JC1,JC2] to the allowed curvature deviation δ.

Once we have constructed $A(R, \delta)$, we can use this algorithm to search for the critical curvature. By the derived relation between the grid-size and δ, we can also give an upper bound on the length of the shortest $(R_c - \delta)$-constrained path.

A more practical application of $A(R, \delta)$ is however an algorithm, which constructs an R-constrained path in time dependent on $W = |R_c - R|/R$. Similar approaches have been successfully applied to traditional motion planning [Al+]. Omitting the dependency on l_{min} and assuming $W < 1$, we obtain the worst case running time

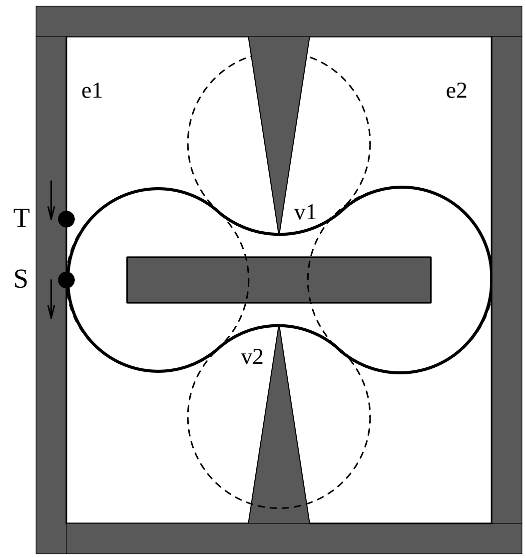

figure 1

$O(|\log W| \cdot (n^2(\frac{R}{W})^2 + n^3(n+\frac{R}{W})\log(n+\frac{R}{W})))$.

Note that, for $W \neq 0$, this provides a decision algorithm which is easier and more practical than the (inherently exponential) solution presented in [FW]. Although our running time can be exponential for exponentially small l_{min} or W, the decision scheme is polynomial in the absolute values of l_{min}^{-1} and W^{-1}.

Section 4 describes and analyses the adequate algorithms on top of $A(R, \delta)$.

2 The Visibility Approach

2.1 Canonical Trajectories

To start with, we review normal form properties of curvature constrained paths.

A *canonical trajectory* with respect to constraint R is a path, which first follows an arc of radius R, then either a straight line or another arc of radius R, and finally again an arc of radius R. This includes degenerate cases, in which the *initial arc*, the *line segment*, or the *final arc* are missing.

For a given configuration $C = (p, u)$, there are two circles of radius R passing through p and tangential to u, corresponding to a left or right hand turn. Given start and target configurations S and T, there are thus at most 6 different canonical trajectories from S to T, corresponding to the *turn types* $\mathcal{T} = \{LSL, LSR, RSL, RSR, LRL, RLR\}$.

Dubins [Du] shows, that the shortest path between S and T is a canonical trajectory provided $E = \mathbf{R}^2$. Jacobs and Canny [JC1,JC2] extend this result to polygonal environments:

Lemma 1 *Among all R-constrained paths w from S to T in E, there exists a path w of minimal length, which touches ∂E only finitely many times, and which is canonical between two contact configurations.*

More precisely, the path w touches ∂E during finitely many time intervals $[t_1, t_2]$. The contact configurations may be defined as those configurations $(w(t_i), \dot{w}(t_i))$, at which w enters or leaves ∂E.

If we drop the requirement on w to be shortest, and assume E to be bounded, then a deformation argument shows that we only need to consider canonical trajectories of one of the types $\mathcal{T}' = \{LSL, LSR, RSL, RSR\}$ (see [La2]).

2.2 Grid Construction

Consider a vertex v, with 2 incident edges e_1 and e_2, emanating from v with direction vectors u_1 and u_2. Interpreting direction vectors as points on the unit circle, let $[u_1, u_2]$ denote the arc between u_1 and u_2 of length $< \pi$. Then the set of possible contact configurations for v is either empty $\mathcal{C}(v) = \emptyset$, or

$$\mathcal{C}(v) = \{v\} \times [u_1, -u_2] \ \cup \ \{v\} \times [u_2, -u_1].$$

To stress the algebraic nature of the problem, we may parameterize $u \in [u_1, u_2]$ by its orthogonal projection onto the line $e = xu_v$, with $x \in [-1, 1]$ and u_v the bisector between u_1 and u_2 (rather than by the angle θ with $u = (\sin\theta, \cos\theta)$).

For an edge $e = \{\ p + xu\ ;\ x \in [0, l]\ \}$, $||u|| = 1$, the possible contact configurations are given by

$$\mathcal{C}(e) = e \times \{u\} \ \cup \ e \times \{-u\}.$$

Each set $\mathcal{C}(f)$ consists of 2 connected components $\mathcal{C}^+(f)$ and $\mathcal{C}^-(f)$, $f \in \{v, e\}$. They constitute 1-dimensional curves in 4-space, parameterized by the real parameter x. We subdivide the curves $\mathcal{C}^\pm(f)$ into segments of *uniformly bounded* Euclidian length $\leq \varepsilon$. Clearly, this can be achieved by $O(1/\varepsilon)$ break points for $\mathcal{C}^\pm(v)$, and $O(l/\varepsilon)$ break points for $\mathcal{C}^\pm(e)$, with l bounded above by the length l_{max} of the longest obstacle edge (the same dependency is present in [JC1,JC2]).

In the following, we shall make implicit use of the parameterization of $\mathcal{C}^\pm(f)$. We assume break points to be given by algebraic numbers $x_i \in \mathbf{R}$, and the midpoint $C_{mid}(\sigma)$ of a segment $\sigma \equiv [x_i, x_{i+1}]$ to be the configuration corresponding to $(x_i + x_{i+1})/2$.

The break points subdivide each $\mathcal{C}^\pm(f)$ into at most

$$M = O((l_{max} + 1)/\varepsilon)$$

segments. $\mathcal{S}$ will denote the collection of all the segments, and $\sigma(S)$ (resp., $\sigma(T)$) the segment containing the start (target) configuration.

2.3 Computing Visibility

We call a segment σ_2 *visible* from σ_1, if there exist configurations $C_1 \in \sigma_1$, $C_2 \in \sigma_2$, such that one of the canonical trajectories from C_1 to C_2 is collision-free. This defines a visibility relation

$R_v \subseteq \mathcal{S} \times \mathcal{S}$, or a *visibility graph* $G_v = (\mathcal{S}, R_v)$. Note that the relation R_v is not symmetric, and that G_v is directed.

Our goal is to compute the visibility graph G_v (or $G_v(R, \varepsilon)$, to stress the dependency of G_v on R and the grid size). This is done in 2 steps, performed for each pair $(\mathcal{C}^{\pm}(f_1), \mathcal{C}^{\pm}(f_2))$, and for each turn type $T \in \mathcal{T}'$:

(1) Compute the free space $\mathcal{F}(T) \subseteq \mathcal{C}^{\pm}(f_1) \times \mathcal{C}^{\pm}(f_2)$, consisting of those configurations that can be connected by a collision-free canonical trajectory of type T.

(2) For any grid rectangle $\sigma_1 \times \sigma_2 \subseteq \mathcal{C}^{\pm}(f_1) \times \mathcal{C}^{\pm}(f_2)$, decide if $(\sigma_1 \times \sigma_2) \cap \mathcal{F}(T)$ is empty.

Finally, $(\sigma_1, \sigma_2) \in R_v$ if and only if $(\sigma_1 \times \sigma_2) \cap \mathcal{F}(T) \neq \emptyset$ for some $T \in \mathcal{T}'$.

Let us analyse step (1). The boundary of $\mathcal{F}(T)$ in parameter space consists of algebraic curves of bounded degree. As shown in [JC1,JC2], they are defined by the following *constraints*: the initial or final arc of the canonical trajectory has two obstacle contacts, the straight line segment touches an obstacle vertex, or the initial and final arcs are tangential.

There are $O(n)$ such constraint curves, intersecting in $O(n^2)$ points. An explicit boundary representation of $\mathcal{F}(T)$ can thus be computed by plane sweep in $O(n^2 \log n)$ time. Important for practical implementation, all event points can be computed by sequences of simple geometric primitives (e.g. to find a circle tangential to a line and passing through a point), see [JC1,JC2].

In step (2), we perform plane sweep over the computed representation of $\mathcal{F}(T)$, to decide for each grid rectangle if it contains a free configuration. This closely follows [CSY].

There are 3 kinds of events for the vertical sweep line:

(i) vertices and extremal points on the boundary of $\mathcal{F}(T)$,

(ii) the break points of $\mathcal{C}^{\pm}(f_1)$, and

(iii) intersections between a horizontal grid line and a boundary arc of $\mathcal{F}(T)$.

The event points of type (iii) are dynamically inserted into the priority queue. Whenever a grid line and a boundary arc become adjacent on the sweep line, we check if they will intersect. This amounts to finding the initial arc of a canonical trajectory under a given constraint when the final arc is fixed, and can again be expressed by simple geometric primitives.

Whenever we come to an event of type (ii), we have completely swept a column of at most M grid rectangles, and check for visibility in $O(M)$ time. In case $(\sigma_1 \times \sigma_2) \cap \mathcal{F}(T) \neq \emptyset$, we may also output a *sample trajectory* realizing the visibility, i.e. a pair of configurations $(C_1, C_2) \in (\sigma_1 \times \sigma_2) \cap \mathcal{F}(T)$.

There are $O(n^2)$ events of type (i), and $O(nM)$ events of type (iii). Each can be processed in $O(\log(n+M))$ time. Thus the sweep has an overall time complexity of $O(M^2 + n^2 \log(n + M) + nM \log(n+M))$.

Summing over all pairs $(\mathcal{C}^{\pm}(f_1), \mathcal{C}^{\pm}(f_2))$, we get:

Lemma 2

(1) The visibility graph $G_v(R, \varepsilon)$ can be computed in time

$$O(n^2M^2 + n^3M\log(n+M) + n^4\log(n+M)).$$

(2) If there exists an R-constrained path from S to T inside E, then $G_v(R, \varepsilon)$ contains a path from $\sigma(S)$ to $\sigma(T)$.

As in [CSY], the algebraic formulation allows us to perform plane sweep in the framework of exact geometric algorithms. This means that operations may be carried out to a precision which is sufficient to guarantee that all comparisons are made error-free. But unlike [CSY], our problem involves algebraic formulas of very high degree, making the exact computation paradigm a merely theoretical approach. All given time complexities are thus understood in an algebraic computation model rather than a bit model.

3 Path Smoothening

Any path $\tilde{w} = (\sigma_0, \ldots, \sigma_k)$ in G_v corresponds to a sequence of sample trajectories t_i, $i = 1 \ldots k$, which connect some configuration $C_{i-1}^{+} \in \sigma_{i-1}$ to $C_i^{-} \in \sigma_i$. Let $C_i^{\pm} = (P_i^{\pm}, u_i^{\pm})$. If σ_i corresponds to a corner contact, then $P_i^{+} = P_i^{-}$ but the smoothness condition $u_i^{+} = u_i^{-}$ may be violated. If σ_i corresponds to a wall contact, then $u_i^{+} = u_i^{-}$ but possibly $P_i^{+} \neq P_i^{-}$.

Let $C_0^m = S$, $C_k^m = T$, and $C_i^m = C_{mid}(\sigma_i)$ for $i = 1 \ldots k-1$. Our goal is to construct a smooth path w, which connects the configurations C_i^m by $(R-\delta)$-constrained paths w_i. The subject of this section is to establish a relationship between ε (the maximal length of each σ_i in parameter space) and δ.

Let us focus on a contact configuration $C_s = (P_s, u_s) \in \sigma$, and a canonical trajectory t starting at C_s. Further, let

$$l_{min} = min\{\ dist(v, e)\ \},$$

where v (resp., e) ranges over all non-adjacent obstacle vertices (edges).

To construct a smooth path w as above, it suffices to show that we can construct a free path t' with the following properties:

(1) t' is $(R-\delta)$-constrained,

(2) t' starts at $C_m = (P_m, u_m) = C_{mid}(\sigma)$, and

(3) t' ends at a configuration $C_f = (P_f, u_f)$ on t, with $||P_s - P_f|| \leq l_{min}/3$.

A simple deformation argument shows, that we can confine ourselves to the case where t consists of 1 or 2 arcs of radius R (but no straight line part). Let thus t be a given 2-arc trajectory from C_s to C_f, with $||P_f - P_s|| = l_{min}/3$.

Our goal is to find the biggest ε (in terms of δ, R and l_{min}), for which an $(R-\delta)$-constrained path t' from C_m to C_f always exists. In the following, we first give explicit constructions for t', and then analyze for which ε these constructions are possible.

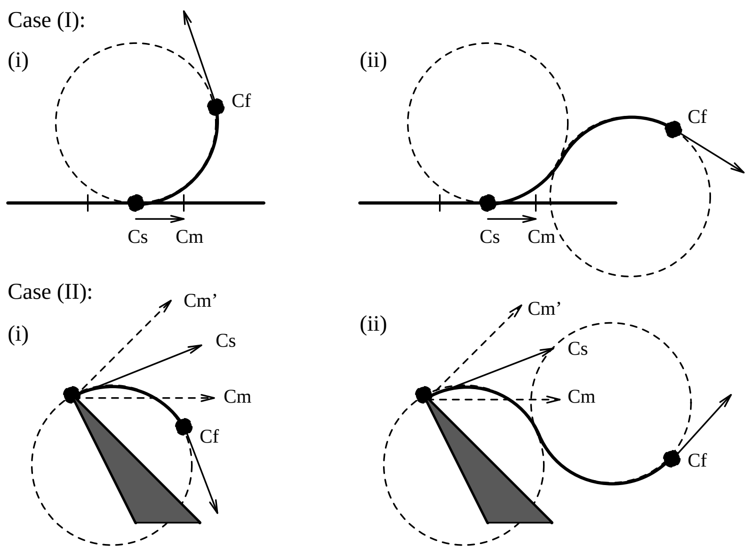

figure 2

Basically, our constructions involve using arcs of curvature $R-\delta$ at either C_m or C_f, and straight line connections between arcs. The constructions are held as simple as possible, in order to allow the required analysis.

With account to symmetry, we get the following possibilities, see figure 2:

(I) C_s is a wall contact:
If P_m lies in negative direction to P_s with respect to u_s, then we can construct t' by expanding t with a straight line connection between C_s and C_m. We thus assume that P_m lies in positive direction to P_s.

(i) t consists of one arc A_s:
The trajectory t' will consist of an initial arc A'_m with radius $R-\delta$ at C_m, a final arc A'_f with radius R at C_f, and a straight line L tangential to A'_m and A'_f.

(ii) t consists of two arcs $A_s A_f$, intersecting at C:
If A_s is longer than A_f, we set $C_f = C$ and proceed as in (i). If A_f is longer than A_s, then t' will consist of an initial arc A'_m with radius R at C_m, a final arc A'_f with radius $R-\delta$ at C_f, and a straight line segment L tangential to both.

(II) C_s is a corner contact:
For the 1-arc and 2-arc trajectories, we have to distinguish if u_m lies clock- or counter-clockwise to u_s, or equivalently, if the first portion of A_s lies between u_s and u_m (a) or not (b). In case (b), the 1-arc (i) and 2-arc (ii) instances can be handled analogous to (I). In case (a), both instances can be reduced to case (b) (we omit the details).

In the following, we review some basic ideas of the technical analysis. Throughout the analysis, we assume

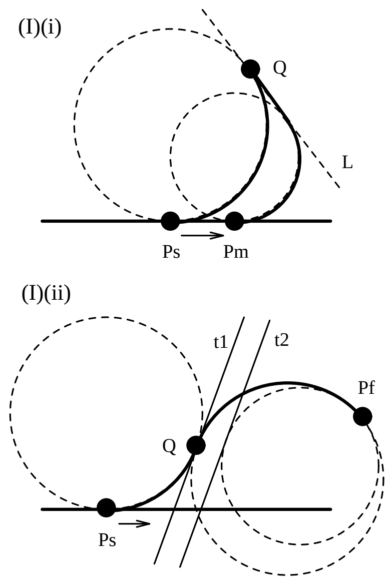

figure 3

$$\delta \leq R/2$$

and

$$l_{min} \leq const \cdot R.$$

(Note that for $l_{min} \geq 4R$, there always consists an R-constrained path provided P_S and P_T lie in the same connected component of E.)

Let us consider case (I)(i) (figure 3):

W.l.o.g. we may assume the wall to be the x-axis, i.e. $u_s = (1, 0)$, and P_s to be the origin, $P_s = (0, 0)$. Let $P_m = (\varepsilon, 0)$, and let Q be the intersection of the line L with the circle of radius R at C_s. The distance between Q and P_s can be calculated as

$$||Q - P_s|| = 2\frac{R\varepsilon}{\sqrt{\varepsilon^2 + \delta^2}}.$$

If we choose

$$\varepsilon \leq \frac{l_{min}}{6R}\delta,$$

then this evaluates to $||Q - P_s|| \leq l_{min}/3$, proving the validity of the construction.

The case (I)(ii) is significantly harder. Indeed, the involved equations already exceed the solving capabilities of common algebra systems, making good estimates a necessity. We consider the case that A_f is longer than A_s (figure 3):

Let Q be the intersection (i.e., common point) of A_s and A_f. As A_f is longer than A_s, we have $||Q - P_f|| \geq l_{min}/6$. Now let t_1 be the tangent to A_s passing through Q, and t_2 the line parallel to t_1 and tangential to the circle with radius $R - \delta$ at C_f. Then the tangent lines t_1 and t_2 intersect the x-axis at I_1 and I_2, with

$$||I_1 - I_2|| = \Omega(\frac{l_{min}}{R}\delta).$$

This means that for

$$\varepsilon = O(\frac{l_{min}}{R}\delta)$$

the arcs A'_m and A'_f do not intersect, proving again the validity of the construction.

The 4 cases in (II) are even more complicated. The upshot of similar estimates as above finally leads us to

$$\varepsilon = O(\frac{l_{min}}{R^2}\delta),$$

which is by the factor $1/R$ worse than in (I). This has a simple explanation, on which our analysis is partly based on: observe that a rotation of a circle with radius R at a configuration C by an angle $\Theta(\varepsilon)$ corresponds to a motion of the midpoint of the circle along an arc of length $\Theta(\varepsilon R)$.

Combining the above, we get:

Lemma 3 *Let t be a canonical trajectory starting at $C_s \in \sigma$, and $C_m = C_{mid}(\sigma)$. If the grid size is bounded by*

$$\varepsilon = \Theta(\frac{l_{min}}{R^2}\delta)$$

then there exists a path t' satisfying conditions (1)-(3).

Corollary 1 *If the visibility graph $G_v(R, \varepsilon)$ contains a path $\tilde{w}$ from $\sigma(S)$ to $\sigma(T)$, and ε is chosen as in lemma 3, then there exists an $(R - \delta)$-constrained path w from S to T in E.*

We conclude this section by an interesting bound on the length of paths with curvature near the critical curvature:

Theorem 1 *Let R_c be the critical curvature with respect to S and T, let D denote the diameter of the (bounded) environment E, and let $L = (l_{max} + 1)/l_{min}$. Then E contains an $(R_c - \delta)$-constrained path from S to T with length*

$$O(D \cdot n^2 \cdot (L\frac{R_c^2}{\delta})^2).$$

Proof (Sketch): By definition of R_c, E contains an $(R_c - \frac{1}{2}\delta)$-constrained path. Now let G'_v be the visibility graph respective to $R' = R_c - \frac{1}{2}\delta$, $\delta' = \frac{1}{2}\delta$, and ε' as in lemma 3. By lemma 2, G'_v will contain a path $\tilde{w}$ from $\sigma(S)$ to $\sigma(T)$. By lemma 3, the path $\tilde{w}$ corresponds to an $(R' - \delta')$-constrained path w in E, with $R' - \delta' = R_c - \delta$. But G'_v and thus $\tilde{w}$ contains at most $O(n^2 M'^2)$ edges, each corresponding to a portion of w with length at most $O(D)$. This proves the claim. □

4 Algorithms

Let $A(R, \delta)$ be the algorithm described in section 2 to compute the visibility graph $G_v(R, \varepsilon)$ – with ε depending on δ as in lemma 3, and extended by a simple graph search to decide whether G_v contains a path from $\sigma(S)$ to $\sigma(T)$.

Combining lemmata 2 and 3 shows:

- If $A(R, \delta)$ outputs "yes", then there exists an $(R - \delta)$-constrained path from S to T in E.
- If $A(R, \delta)$ outputs "no", then there exists no R-constrained path from S to T in E.

Let $L = (l_{max} + 1)/l_{min}$. With lemma 2, the running time of $A(R, \delta)$ computes to

$$O(n^2 L^2 \frac{R^4}{\delta^2} + n^3 (L \frac{R^2}{\delta} + n) \log(L \frac{R^2}{\delta} + n)).$$

By running the algorithm $A(R, \delta)$ for different values of R and δ, we will construct an algorithm to compute the critical curvature, and a decision algorithm for fixed R.

Note that for $\delta = const \cdot R$, the dependency of running times on R and $1/l_{min}$ is the same: this shows that scaling down E and R simultaneously does not affect running times.

For brevity of description, we will assume l_{min} and l_{max} as constant in the sequel.

4.1 Computing the Critical Curvature

Our goal is to find, for any given $\varepsilon > 0$, a value R'_c with $|R_c - R'_c| \le \varepsilon R_c$.

Note that, if the query $A(R, \delta)$ returns "yes", we know that there exists an R'-constrained path for any $R' \in [0, R - \delta]$, and thus $R_c \ge R - \delta$. If $A(R, \delta)$ returns "no", then there exists no R'-constrained path for any $R' \in [R, \infty]$, and thus $R_c \le R$.

The algorithm A1, described below, first computes an upper bound R_m on R_c, and then successively refines the interval $[a, b] \subseteq [0, R_m]$. The correctness directly follows from

(1) $R_m/4 \le R_c \le R_m$, and

(2) $R_c \in [a, b]$.

Algorithm A1

(I) Find R_m s.t. $R_c \in [R_m/4, R_m]$:

(1) $R := 2$; $\delta := 1$;
(2) If $A(R, \delta) =$ "yes"
(3) Then Repeat $R := 2R$; $\delta := 2\delta$;
(4) Until $A(R, \delta) =$ "no";
(5) $R_m := R$;
(6) Else Repeat $R := R/2$; $\delta := \delta/2$;
(7) Until $A(R, \delta) =$ "yes";
(8) $R_m := R/2$;
(9) Fi.

(II) Refine $[a, b] \subseteq [0, R_m]$:

(1) $a := 0$; $b := R_m$;
(2) Repeat $m := a + (b - a)/2$; $\delta := (b - a)/4$;
(3) If $A(m, \delta) =$ "no"
(4) Then $b := m$;
(5) Else If $A(m + \delta, \delta) =$ "yes"
(6) Then $a := m$;
(7) Else $a := a + \delta$; $b := b - \delta$;
(8) Fi;
(9) Fi;
(10) Until $(b - a) < \varepsilon R_m$;
(11) $R'_c := a$.

Let us analyse the time complexity of part (I).
If the query $A(R, R/2)$ in line (2) has result "yes", then $R_c \geq 1$ and we successively double R until $A(R, R/2) =$ "no". But this means $A(R/2, R/4) =$ "yes", and hence $R_c \geq R/4$. The query time for each call of A is

$$O(n^2 R_c^2 + n^3 (R_c + n) \log(R_c + n)).$$

In case the query in line (2) has result "no", then $R_c \leq 2$ and we successively half R. Using $R \leq 2$, the query time computes to $O(n^4 \log n)$.
The number of queries in each case is $O(|\log R_c|)$.

In part (II), we half in each loop the size of the interval $I = [a, b]$. In the i-th loop, we have $\delta = \Theta(2^{-i} R_c)$. The program exits the loop in line (10) if $4\delta < 4\varepsilon R_c$. Hence $\delta \geq \varepsilon R_c$ in each call of $A(R, \delta)$. Further, $R \leq 4R_c$ in each such call. Thus each query (loop) has time complexity

$$O(n^2 (\frac{R_c}{\varepsilon})^2 + n^3 (n + \frac{R_c}{\varepsilon}) \log(n + \frac{R_c}{\varepsilon})).$$

The number of queries (loops) is $i = O(|\log \varepsilon|)$.

Summing up, we get:

Theorem 2 *The critical curvature R_c can be approximated with relative error ε in time*

$$O((|\log \varepsilon| + |\log R_c|) \cdot$$
$$(n^2 (\frac{R_c}{\varepsilon})^2 + n^3 (n + \frac{R_c}{\varepsilon}) \log(n + \frac{R_c}{\varepsilon}))).$$

An explicit representation of an R'_c-constrained path can be computed in the same time bound.

4.2 A Width-Dependent Decision Algorithm

Let R be a given curvature. Our goal is to decide, if there exists an R-constrained path from S to T.

Let $\delta \leq R/2$. Then the desired decision can be made if either $A(R + \delta, \delta)$ has output "yes", or if $A(R, \delta)$ has output "no". If both attempts fail, then $R_c \in I_\delta = [R - \delta, R + \delta]$, and we repeat the queries for $\delta := \delta/2$. This defines a loop which will terminate once $R_c \notin I_\delta$.

Thus the running time of the algorithm A2, described below, depends on the relative width $W = |R - R_c|/R$. The decision is fast if W is large. However if $W = 0$, the program will not terminate.

Algorithm A2

```
δ := R/2;
Repeat
If      A(R + δ, δ) = "yes"
Then    Print("yes"); Exit;
Else    If      A(R, δ) = "no"
        Then    Print("no"); Exit;
        Fi;
Fi;
δ := δ/2;
Until   False.
```

In the i-th loop, $\delta = \Theta(2^{-i}R)$. The length of the interval I_δ equals 2δ. The program exits the loop if $\delta < |R - R_c|/2$, or $2\delta < WR$. If $W \geq 1$, there will only be a constant number of queries (loops), each with $\delta = \Theta(R)$, and time complexity

$$O(n^2R^2 + n^3(n + R)\log(n + R)).$$

If $W < 1$, then the number of loops equals $i = |\log W|$, and $\delta = \Omega(WR)$ in each query to A. The query time is

$$O(n^2(\frac{R}{W})^2 + n^3(n + \frac{R}{W})\log(n + \frac{R}{W})).$$

Summing up, we get:

Theorem 3 *Let $W' = min\{W, 1\}$. Given $R > 0$, the existence of an R-constrained path can be decided in time*

$$O(\ (|\log W'| + 1) \\ (n^2(\frac{R}{W'})^2 + n^3(n + \frac{R}{W'})\log(n + \frac{R}{W'}))).$$

Again, this result includes the construction of an R-constrained path if the decision question is answered positive.

References

[Al+] H. Alt, R. Fleischer, M. Kaufmann, K. Mehlhorn, S. Näher, S. Schirra, C. Uhrig, "Approximate Motion Planning and the Complexity of the Boundary of the Union of Simple Geometric Figures".

[BCL] J. Boissonnat, A. Cérézo, J. Leblond, "Shortest paths of bounded curvature in the plane", Technical Report, INRIA Sophia Antipolis.

[CSY] J. Choi, J. Sellen, C.-K. Yap, "Approximate Euclidian Shortest Path in 3-Space", *Proc. 10th ACM Symposium on Computational Geometry*, 1994, pp. 41-48.

[Du] L.E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents", *American Journal of Mathematics 79*, 1957, pp. 497-516.

[FW] S. Fortune, G. Wilfong, "Planning Constrained Motion", *Annals of Mathematics and AI 3*, 1991, pp. 21-82.

[JC1] P. Jacobs, J. Canny, "Planning Smooth Paths for Mobile Robots", *IEEE Conference on Robotics and Automation*, 1989, pp. 2-7.

[JC2] P. Jacobs, J. Canny, "Planning Smooth Paths for Mobile Robots", in: *Nonholonomic Motion Planning*, ed. Z. Li, J. Canny, Kluwer Academic Press 1993, pp. 271-342.

[La1] J. Laumond, "Finding collision-free smooth trajectories for a non-holonomic mobile robot", *IJCAI 10*, 1987, pp. 1120-1123.

[La2] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.

[Pa] C.H. Papadimitriou, "An Algorithm for Shortest-Path Motion in Three Dimensions", *Information Processing Letters*, Vol. 20, 1985, pp. 259-263.

[RS] J.A. Reeds, L.A. Shepp, "Optimal paths for a car that goes both forwards and backwards", *Pacific Journal of Mathematics 145(2)*, 1990, pp. 367-393.