Soft Subdivision Planner for a Rod*

Ching-Hsiang Hsu† Yi-Jen Chiang‡ Chee Yap§

In this paper we consider the motion planning problem to find a path for a rod (also known as a line segment or ladder) amidst polyhedral obstacles in 3D. The configuration space of a rod has 5 degrees of freedom (DOFs) and may be identified with $\mathbb{R}^3 \times S^2$. Devising theoretically-sound but practical algorithms for motion planning of robots with more than 4 DOFs is a major challenge. To our knowledge, there is no explicit exact planning algorithm for a rod. Even if one exists, exact implementation seems out of the question. We also believe that exact algorithms have little practical value in robot motion planning because of the inherent inaccuracies in all real-world robots and environments. So we want ε-approximations, but we must be careful not to use the ε parameter as if it is an exact parameter (e.g., replacing a point p by an ε-ball centered at p with sharp boundaries). In our recent work [3, 5, 6, 1], we propose the notion of a resolution-exact algorithm where ε is used in a "soft" manner. Specifically, we propose a subdivision framework based on the twin foundations of *ε-exactness* and *soft predicates*. In this paper we develop an ε-exact planner for a rod in 3D under this framework.

Technical Background and Setting

We consider a robot R_0 which is a **rod** (closed line segment) of length $\ell_0 > 0$ in $\mathbb{R}^3$. The two endpoints of the rod are denoted B and A, and called (respectively) the **base** and **apex** of R_0. The **standard placement** of the rod is defined to be the line segment $[(0,0,0),(0,0,\ell_0)]$ where $(0,0,0)$ and $(0,0,\ell_0)$ correspond to the placements of B and A, respectively.

As a rigid body, the rod has 5 DOFs. We let its configuration space be $C_{space} = \mathbb{R}^3 \times S^2$ and a typical configuration is a pair (q,α) where $q = (x,y,z) \in \mathbb{R}^3$ and $\alpha \in S^2$ is a (rod) **orientation**. This configuration defines a transformation of $\mathbb{R}^3$ whereby the base and apex of the robot are respectively transformed to the points $B[q,\alpha] := q$ and $A[q,\alpha] := q + \ell_0\alpha$.

The space S^2 is unsuitable for subdivision. So we introduce a parametrized model $\widehat{S^2}$ which is the boundary of the cube $[-1,1]^3 \subseteq \mathbb{R}^3$. In general, if X and Y are metric spaces, we call Y a **good parametrization** of X if there is a constant $K > 1$ and homeomorphism $h : Y \to X$ such that for all $x, y \in Y$, $\frac{\|x-y\|}{\|h(x)-h(y)\|} \in [1/K, K]$. Here, $\|.\|$ denotes the metric in the respective space. It is easy to see that $\widehat{S^2}$ is a good parametrization of S^2, called the **square model** (as given in [6] in a more general framework) of the 2-sphere S^2. There is a canonical map $\widehat{\cdot} : \widehat{S^2} \to S^2$ where $\widehat{\alpha} \mapsto \alpha = \frac{\widehat{\alpha}}{\|\widehat{\alpha}\|}$. For $v \in \{x,y,z\}$, we call $+v$ or $-v$ a **direction**. So there are 6 directions and they identify the faces of $\widehat{S^2}$ as follows: $\widehat{S^2}_{+v}$ (resp., $\widehat{S^2}_{-v}$) denotes the face of $\widehat{S^2}$ in which $v = 1$ (resp., $v = -1$). For instance, $\widehat{S^2}_{+x} = \{(1,y,z) : -1 \le y,z \le 1\}$ and $\widehat{S^2}_{-z} = \{(x,y,-1) : -1 \le x,y \le 1\}$.

Fix some set $\Omega \subseteq \mathbb{R}^3$ of obstacles. The **footprint** of the rod AB at configuration $\gamma = (q,\alpha) \in C_{space}$ is the closed line segment with the base at $B[q,\alpha]$ and apex at $A[q,\alpha]$. Let $AB[\gamma]$ denote this footprint. The **separation** between two sets $S,T \subseteq \mathbb{R}^3$ is $\mathrm{Sep}(S,T) = \inf\{\|s-t\| : s \in S, t \in T\}$. When S is fixed, we also write $\mathrm{Sep}_S : \mathbb{R}^3 \to \mathbb{R}_{\ge 0}$ where $\mathrm{Sep}_S(q) = \mathrm{Sep}(S,\{q\})$. The **$\Omega$-clearance** function $C\ell_\Omega : C_{space} \to \mathbb{R}_{\ge 0}$ is given by $C\ell_\Omega(\gamma) := \mathrm{Sep}(AB[\gamma],\Omega)$. We omit the subscript Ω when it is understood. We say γ is **free** if $C\ell(\gamma) > 0$.

For any non-zero point $q = (x,y,z) \in \mathbb{R}^3$, let $\widehat{q}$ denote the point

$$\widehat{q} := (x/d, y/d, z/d)$$

where $d = \max\{|x|,|y|,|z|\}$. Note that $\widehat{q} \in \widehat{S^2}$. For instance, if $q = (-2,1,0)$ then $\widehat{q} = (-1,0.5,0) \in \widehat{S^2}_{-x}$.

We say a planner is **resolution-exact** if there is an "accuracy" constant $K > 1$ such that for all input instances and any input "resolution" parameter $\varepsilon > 0$, if there is a path with clearance $K\varepsilon$, the planner must output a path[1], and if there is no path of clearance ε/K, the planner must output NO-PATH. Note that if the maximum clearance of the input instance is in the range $[\varepsilon/K, K\varepsilon)$, the planner may either return a path or say NO-PATH. The use of ε-exact algorithms has some major consequence: fundamentally, this avoids exact computation (i.e., the Zero Problem). First, it means we can use numerical approximations to solve our

*This work is supported by the National Science Foundation under Grant No. CCF-1423228.

†Department of Computer Science, New York University, New York, NY; email: chhsu@nyu.edu.

‡Department of Computer Science and Engineering, New York University, Brooklyn, NY; email: chiang@nyu.edu.

§Department of Computer Science, New York University, New York, NY; email: yap@cs.nyu.edu.

[1] Note that we do not require that the output path has any particular clearance. A variation is to require that the output path has clearance ε/K (i.e., if there is a path with clearance $K\varepsilon$, the planner must output a path with clearance ε/K). This variation is used in [3].

problem. In particular, non-algebraic problems are now accessible by such algorithms. Second, we can avoid all degeneracy determinations in our algorithms. Note that except for relatively simple (linear) problems, most problems in 3D computational geometry have formidable non-degeneracies for which we currently have no analysis [4]. Thus, exact (and explicit) algorithms are unavailable for such a basic problem as Voronoi diagram of a set of polyhedral objects.

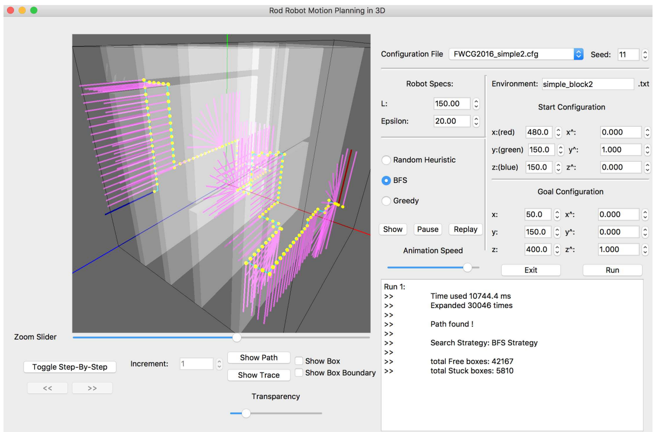

Figure 1: This example environment contains four blocks of "walls", each with a horizontal passage (the first and third walls from left) or a vertical passage (the second and fourth walls). The initial (red) and final (blue) configurations of the rod (with length 150) are at the right and left sides of the walls. The resulting path is shown in cyan, with the trace (i.e., the orientations of the rod at various places) along the path shown in magenta.

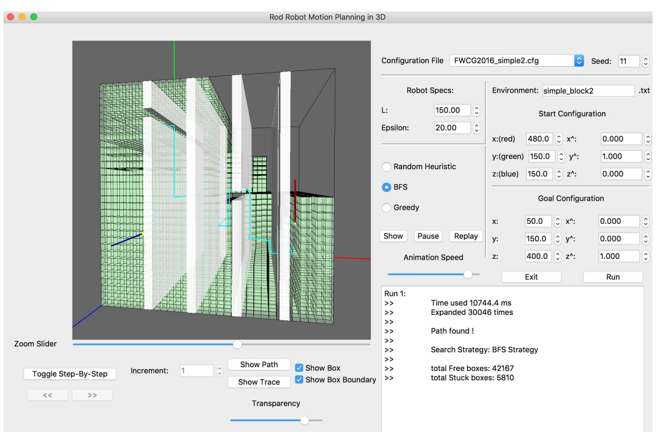

Figure 2: Partial subdivision result of the example environment in Fig. 1. The green boxes mean that they are in the free configuration space for the rod. Again, the initial and final configurations of the rod are respectively shown in red and blue, and the path is shown in cyan.

Summary of Contributions

In this paper we develop an ε-exact planner for a rod in 3D under the general framework of subdivision motion planning [3, 5, 6, 1]. We make three major contributions:

(A) Soft predicates for a rod robot in 3D. As envisioned in [3, 1], soft predicates can exploit a wide variety of techniques that trade-off ease of implementation against efficiency. Here we introduce the notions of the **forbidden orientations** of a **corner feature**, of an **edge feature**, and to a **wall feature**, respectively, in the context of a 3D rod robot.

(B) A subdivision scheme for $\mathbb{R}^3 \times S^2$. Naive subdivision of this space is impractical; we devise a splitting strategy to achieve real-time performance, by extending the "T/R Splitting" technique of [1] for planar 2-link robots (with configuration space $\mathbb{R}^2 \times (S^1)^2$, 4 DOFs) to $\mathbb{R}^3 \times S^2$.

(C) Implementation of the above techniques. Our experiments on a variety of challenging obstacle environments confirm the practicality of our planner. Unlike most practical planners today, we do not use randomization and yet offer much stronger theoretical guarantees of performance.

We have implemented in C++ our motion planner for a rod robot in 3D. Our code and datasets will be freely distributed with the `Core Library`. An example is shown in Figs. 1 and 2, for which our planner can find solution paths efficiently in real time. A video clip showing the animation of a resulting path in more detail is available[2].

References

[1] Z. Luo, Y.-J. Chiang, J.-M. Lien, and C. Yap. Resolution exact algorithms for link robots. In *Proc. 11th Intl. Workshop on Algorithmic Foundations of Robotics (WAFR '14)*, pages 353–370, 2014.

[2] J. Selig. *Geometric Fundamentals of Robotics*. Springer, second edition, 2005.

[3] C. Wang, Y.-J. Chiang, and C. Yap. On Soft Predicates in Subdivision Motion Planning. *Comput. Geometry: Theory and Appl. (Special Issue for SoCG'13)*, 48(8):589–605, 2015.

[4] C. Yap, V. Sharma, and J.-M. Lien. Towards Exact Numerical Voronoi diagrams. In *9th Int'l Symp. of Voronoi Diagrams in Science and Engineering (ISVD)*, pages 2–16. IEEE, 2012. Invited Talk.

[5] C. K. Yap. Soft Subdivision Search in Motion Planning. In A. Aladren et al., editor, *Proc. Workshop on Robotics Challenge and Vision (RCV 2013)*, 2013. A Computing Community Consortium (CCC) **Best Paper Award**, Robotics Science and Systems Conference (RSS 2013), Berlin. In arXiv:1402.3213.

[6] C. K. Yap. Soft Subdivision Search and Motion Planning, II: Axiomatics. In *Frontiers in Algorithmics*, volume 9130 of *Lecture Notes in Comp.Sci.*, pages 7–22. Springer, 2015. Plenary Talk at 9th FAW.

[2] `https://youtu.be/PnlErBRun_s`.