Pianos are not Flat:
Rigid Motion Planning in Three Dimensions

Vladlen Koltun*

Abstract

Consider a robot $\mathcal{R}$ that is either a line segment or the Minkowski sum of a line segment and a 3-ball, and a set $\mathcal{S}$ of polyhedral obstacles with a total of n vertices in $\mathbb{R}^3$. We design near-optimal exact algorithms for planning the motion of $\mathcal{R}$ among $\mathcal{S}$ when $\mathcal{R}$ is allowed to translate and rotate. Specifically, we can preprocess $\mathcal{S}$ in time $O(n^{4+\varepsilon})$ for any $\varepsilon > 0$ into a data structure that given two placements α and β of $\mathcal{R}$, can decide in time $O(\log n)$ whether a collision-free rigid motion of $\mathcal{R}$ between α and β exists and if so, output such a motion in time asymptotically proportional to its complexity. Furthermore, we can find in time $O(n^{4+\varepsilon})$ for any $\varepsilon > 0$ the largest placement of a similar (translated, rotated and scaled) copy of $\mathcal{R}$ that does not intersect $\mathcal{S}$. A number of additional stronger results are provided. Our line segment motion planning algorithm improves the result of Ke and O'Rourke by two orders of magnitude and almost matches their lower bound, thus settling a classical motion planning problem first considered by Schwartz and Sharir in 1984. This implies a number of natural directions for future work concerning rigid motion planning in three dimensions.

1 Introduction

Background. Algorithmic robot motion planning has been intensively studied for over two decades. We consider exact algorithms for planning collision-avoiding motions for holonomic robots among rigid obstacles, where the motion is not required to be optimal. We invite the reader to consult the many available books and surveys [17, 41] for information on other problem variants.

General approaches to robot motion planning were developed in the early 80s [32, 39]. One of the first concrete settings for which near-optimal algorithms were derived was the case of a line segment robot (referred to in the literature as a *rod* or *ladder*, and in this paper as a *needle*) moving rigidly (i.e., translating and rotating) among polygonal obstacles of overall combinatorial complexity (henceforth, just *complexity*) n in the plane. Following a sequence of results [29, 34, 35, 44], Vegter introduced an optimal $\Theta(n^2)$ algorithm for the problem [36, 46]. Needle motion planning techniques were generalized to planning the motion of a convex polygonal robot of complexity k and after a string of improvements [13, 24, 25, 30], a near-$O((nk)^2)$ algorithm was presented [3]. Using different ideas, near-$O((nk)^2)$ algorithms were also derived for non-convex robots of complexity k [18, 19]. (See [8] for early work on this setting.) The problem was also studied when the robot and/or the obstacles satisfy other special geometric properties, see, e.g., [45].

The problem becomes much simpler if the robot is only allowed to translate in the plane, without changing its orientation. The structure of this motion planning problem is well understood [4, 15, 21, 23, 31] and it can be solved in very general settings with robots of complexity k translating within a polygonal (or more general curved) environment of complexity n in time close to $O(nk)$ [15].

It is this simpler variant of the problem that only allows translational motion that has been most studied in three dimensions, with notable success. Near-quadratic (in the overall complexity of the obstacles times the complexity of the robot) algorithms have been developed for planning translational motion amidst polyhedral obstacles in three dimensions of a box or a convex polygon [20], a convex polyhedron [7] or a ball [6].

Rigid motion planning, when the robot is allowed to translate and rotate, is arguably more natural than purely-translational motion planning.[1] However, it is also considerably more involved, due to the increase in the number of degrees of freedom of the robot in three dimensions, from three in the purely translational case

*Computer Science Division, University of California, Berkeley, CA 94720-1776, USA; *vladlen@cs.berkeley.edu*; *http://www.cs.berkeley.edu/~vladlen/*.This work has been partially supported by NSF Grant CCR-01-21555.

[1] The reader is invited to recall the last time they had to maneuver a bulky furniture item through a stairwell when relocating; after all, it is this setting that gave the "piano movers" problem its name [39]. The title of the current paper refers to the present situation in which algorithms for rigid motion planning have been studied primarily in the plane.

to five or six. As in the plane, the simplest instance of rigid motion planning in 3-space is that of a needle among polyhedral obstacles of overall complexity n. This setting was considered by Schwartz and Sharir, who gave an $O(n^{11})$ algorithm [40] that has been improved by Ke and O'Rourke [22] to $O(n^6 \log n)$. Ke and O'Rourke also provided a strong lower bound for this problem, constructing an example in which the simplest motion of the needle from one placement to another has $\Omega(n^4)$ steps.

The roadmap algorithm developed by Canny [9, 10, 11] solves the robot motion planning problem in very general settings in time close to $O(n^d)$, where d is the number of degrees of freedom. This approach does not utilize the geometric structure present in many natural classes of motion planning problems and often does not result in optimal solutions. In the case of needle motion planning in 3-space, it runs in time super-$O(n^5)$, which is an order of magnitude away from the lower bound [22]. The problem of devising a near-optimal needle motion planning algorithm in $\mathbb{R}^3$ has been repeatedly considered since the late 80s [16]. Despite its appeal and recognition, no further progress has been made prior to the current work.

Contribution. In this paper we show that the above needle motion planning problem can be solved in near-optimal time. Let $\mathcal{L}$ be a needle and let $\mathcal{S}$ be a set of pairwise disjoint polyhedral obstacles with a total of n vertices in $\mathbb{R}^3$. We show that the complexity of the *free configuration space* [41] of $\mathcal{L}$ among $\mathcal{S}$ is $\Theta(n^4)$. This result holds in more generality for $\mathcal{S}$ being a collection of n pairwise disjoint semi-algebraic sets of constant descriptive complexity. We provide an algorithm that preprocesses this configuration space in time $O(n^{4+\varepsilon})$ into a data structure[2] that given two placements α and β of $\mathcal{L}$ in $\mathbb{R}^3$ can decide in time $O(\log n)$ whether a collision-free rigid motion of $\mathcal{L}$ between α and β exists and if so, output such a motion in time asymptotically proportional to its complexity (which is $\Theta(n^4)$). The motion is not required to be optimal.

Furthermore, consider the problem of finding the largest similar (i.e., translated, rotated and scaled) copy of $\mathcal{L}$ that does not intersect $\mathcal{S}$. We show that it can be solved in time $O(n^{4+\varepsilon})$. This extremal placement problem was previously investigated in two dimensions [2, 3, 12, 13, 43]. Our result is the first such result in three dimensions.

Finally, let a robot $\mathcal{M}$ be the Minkowski sum of a line segment with a 3-dimensional ball (referred to below as a *cigar* following [6]). We show that the complexity of the free configuration space of $\mathcal{M}$ among $\mathcal{S}$ is $O(n^{4+\varepsilon})$ and that the above algorithms can be made to work in this setting without compromising asymptotic performance.

These results point to an abundance of open problems in the area of efficient rigid-motion planning in three dimensions. It is our hope that this research will pave the way for a flourishing in this area, in analogy to the field of planar rigid-motion planning that blossomed following the discovery of near-optimal planar needle motion planning algorithms.

In the next section we introduce notation and preliminary observations that are used throughout the paper. In Section 3 we analyze the complexity of the free configuration space of a needle or a cigar among polyhedral obstacles. In Section 4 we describe the motion planning algorithms and Section 5 outlines the extremal placement algorithms. We conclude in Section 6 with directions for future research.

2 Preliminaries

Let $\mathcal{L} = \mathcal{L}(d)$ be a line segment of length d and let $\mathcal{M} = \mathcal{M}(d, b) = \mathcal{L} \oplus \mathcal{B}$ be the Minkowski sum of $\mathcal{L}$ with a 3-ball $\mathcal{B}$ of radius b.[3] $\mathcal{L}$ is said to be the *backbone* of $\mathcal{M}$. $\mathcal{M}$ comprises two half-ball *caps* $\mathcal{B}_1(\mathcal{M})$ and $\mathcal{B}_2(\mathcal{M})$ of radius b and a cylindrical *torso* $\mathcal{T}(\mathcal{M})$ of radius b and height d. Let $\mathcal{S}$ be a set of pairwise disjoint polyhedral obstacles with a total of n vertices in $\mathbb{R}^3$.

Observation 1. *It is sufficient throughout this paper to consider $\mathcal{S}$ to be a collection $\{s_1, \ldots, s_n\}$ of n closed pairwise interior-disjoint triangles.*

Proof: Follows since the boundary of a polyhedron P can be decomposed into $O(m)$ triangles, where m is the number of vertices of P. $\square$

Furthermore, throughout the paper we assume for simplicity that the triangles of $\mathcal{S}$ are in general position with respect to each other and $\mathcal{L}$, $\mathcal{M}$. This means, in particular, that $\mathcal{L}$ or $\mathcal{M}$ cannot concurrently touch the boundaries of more than five triangles (since $\mathcal{L}$ and $\mathcal{M}$ have only five degrees of freedom), and the *interior* of $\mathcal{L}$ (or the boundary of the torso of $\mathcal{M}$) cannot touch more than four such boundaries (since a line in $\mathbb{R}^3$ has four degrees of freedom). General position also means in our setting that two triangles of $\mathcal{S}$ are disjoint. All the proofs and algorithms in this paper can be easily extended to correctly apply without the general position assumption.

A *placement* of $\mathcal{L}$ (resp., $\mathcal{M}$) is a congruent copy of $\mathcal{L}$ (resp., $\mathcal{M}$) in $\mathbb{R}^3$. Such a placement is uniquely

[2] Throughout this paper, a complexity bound of the form $f(n) = O(n^{q+\varepsilon})$ means that, for any arbitrarily small positive ε, there exists a constant c_ε, such that $f(n) \le c_\varepsilon n^{q+\varepsilon}$. Generally, $c_\varepsilon \to \infty$ as $\varepsilon \downarrow 0$.

[3] The Minkowski sum $A \oplus B$ of two sets A and B is defined as $A \oplus B := \bigcup_{\alpha \in A} \bigcup_{\beta \in B} \{\alpha + \beta\}$.

determined by the line ℓ containing $\mathcal{L}$ and a real parameter that determines the position of $\mathcal{L}$ within ℓ. (Recall that by our choice of notation, $\mathcal{L}$ also denotes the backbone of $\mathcal{M}$ and a placement of $\mathcal{L}$ determines the placement of $\mathcal{M}$.) Consider the four-dimensional space $\mathbb{L}$ of lines ℓ in $\mathbb{R}^3$, which is homeomorphic to the Grassmanian manifold, a four-dimensional quadric in the five-dimensional projective space $\mathbb{P}^5$. By leaving out the three-dimensional subspace of horizontal lines, we can map the space of nonhorizontal lines homeomorphically to $\mathbb{R}^4$ [38]. The space $\mathcal{C} = \mathcal{C}(\mathcal{L}, \mathcal{S})$ (resp., $\mathcal{C} = \mathcal{C}(\mathcal{M}, \mathcal{S})$) of nonhorizontal placements of $\mathcal{L}$ (resp., $\mathcal{M}$) is isomorphic to $\mathbb{L} \times \mathbb{R}$ and can thus be embedded in $\mathbb{R}^5$. We establish the convention that four coordinate axes x_1, x_2, x_3, x_4 of this space parameterize the line ℓ, while the "vertical" axis x_5 parameterizes the position of $\mathcal{L}$ along ℓ. $\mathcal{C}$ is called the *configuration space* of $\mathcal{L}$ (resp., $\mathcal{M}$). By a sufficiently generic choice of the coordinate system in $\mathbb{R}^3$ we can ensure that leaving out horizontal lines does not affect the validity of the complexity analysis in Section 3 or the correctness of the algorithms in Section 5. The algorithms in Section 4 need to be augmented to account for horizontal placements of the robot during motion planning. We omit these simple adjustments from the extended abstract.

Throughout the paper we speak of placements of $\mathcal{L}$ (resp., $\mathcal{M}$) and corresponding points of $\mathcal{C}$ interchangeably. For an obstacle $s \in \mathcal{S}$, let the corresponding $\mathcal{C}$-*obstacle* s^* be the set of placements of $\mathcal{L}$ (resp., $\mathcal{M}$) that transversally intersect s. Every placement in the union $\bigcup_{s\in\mathcal{S}} s^*$ is said to be *forbidden*, while a placement in the complement of the union is *free*. This complement $\mathcal{F} = \mathcal{C} \setminus \bigcup_{s\in\mathcal{S}} s^*$ is called the *free configuration space*. Its complexity is the overall number of faces of all dimensions on the boundary $\partial\big(\bigcup_{s\in\mathcal{S}} s^*\big)$ of $\mathcal{F}$. (Note that the definitions of $\mathcal{C}$-obstacles and the free configuration space are the same for $\mathcal{L}$ and $\mathcal{M}$, but the obstacles and spaces are different. We distinguish between $\mathcal{F} = \mathcal{F}(\mathcal{L}, \mathcal{S})$ for $\mathcal{L}$ and $\mathcal{F} = \mathcal{F}(\mathcal{M}, \mathcal{S})$ for $\mathcal{M}$.)

Throughout this extended abstract we leave some claims unsubstantiated and omit some details due to space limitations, while trying to convey the main ideas as space allows.

3 Complexity of the Free Configuration Space

3.1 Needles

THEOREM 3.1. *The complexity of $\mathcal{F} = \mathcal{F}(\mathcal{L}, \mathcal{S})$ is $\Theta(n^4)$.*

Proof: The complexity of $\mathcal{F}$ can be asymptotically bounded by estimating the number of vertices of $\mathcal{F}$. Indeed, every face of $\mathcal{F}$ that is incident to a vertex can be charged to such a vertex. In general position, a vertex is charged $O(1)$ times. A face that is not incident to a vertex is a complete connected component of an intersection of at most four $\mathcal{C}$-obstacle boundaries. Since every such intersection has $O(1)$ connected components we can charge a face as above to the (≤ 4)-tuple of $\mathcal{C}$-obstacles defining it, obtaining a bound of $O(n^4)$ on the number of faces of $\mathcal{F}$ not incident to a vertex. We thus concentrate on bounding the number of vertices of $\mathcal{F}$.

OBSERVATION 2. *The number of vertices of $\mathcal{F}$ that correspond to free placements of $\mathcal{L}$ in which the interior of $\mathcal{L}$ touches no obstacle edges is $O(n^2)$.*

Proof: Consider a vertex v of $\mathcal{F}$ that corresponds to a placement of $\mathcal{L}$ in which the interior of $\mathcal{L}$ touches no obstacle edges. Since the obstacles are pairwise disjoint, an endpoint of $\mathcal{L}$ can be adjacent to at most one obstacle. Thus v is contained in at most two $\mathcal{C}$-obstacle boundaries. Since there is a linear number of $\mathcal{C}$-obstacles, and each has constant descriptive complexity, the number of vertices v of this kind is $O(n^2)$. □

We now analyze the number of vertices of $\mathcal{F}$ that correspond to free placements of $\mathcal{L}$ in which the interior of $\mathcal{L}$ touches at least one edge of a triangle of $\mathcal{S}$. Fix such an edge e and consider the space of lines stabbing e. It is a three-dimensional subspace of $\mathbb{L}$, denoted by $\mathbb{L}_e$. Consider the space $\mathcal{C}_e := \mathbb{L}_e \times \mathbb{R}$, comprising all placements of $\mathcal{L}$ that are contained in lines that stab e. Let x_4 denote the "vertical" coordinate axis in $\mathcal{C}_e$ that parameterizes the placement of $\mathcal{L}$ inside any line of $\mathbb{L}_e$, such that a higher value of x_4, for fixed values of x_1, x_2 and x_3 corresponds to a higher placement of $\mathcal{L}$ in $\mathbb{R}^3$. (Recall that we do not consider horizontal placements of $\mathcal{L}$ here.) Let $x_4 = 0$ be the parameter that places $\mathcal{L}$ so that its lower endpoint (in $\mathbb{R}^3$) is incident to e. (This notation does not take into account placements of $\mathcal{L}$ such that $\mathcal{L} \cap e$ is a line segment, for which we set $x_4 = 0$ so that it places the lower endpoint of $\mathcal{L}$ to be incident to an arbitrary point within e. It is easy to show that the number of vertices of $\mathcal{F}$ that correspond to such placements is $O(n)$.)

Note that $\mathbb{L}_e \subseteq \mathbb{L}$ and $\mathcal{C}_e \subseteq \mathcal{C}$. Given a $\mathcal{C}$-obstacle s^*, we refer to $s^* \cap \mathcal{C}_e$ as the corresponding $\mathcal{C}_e$-*obstacle*. For every $\mathcal{C}_e$-obstacle o, consider the sets

$$\begin{aligned} o^- := \{(\alpha,\beta,\gamma,\delta) \quad | \quad & (\alpha,\beta,\gamma) \in \mathbb{L}_e, \\ & \delta = \operatorname{argmax}_\lambda\{(\alpha,\beta,\gamma,\lambda) \in o\}, \\ & \delta < 0\}, \\ o^+ := \{(\alpha,\beta,\gamma,\delta) \quad | \quad & (\alpha,\beta,\gamma) \in \mathbb{L}_e, \\ & \delta = \operatorname{argmin}_\lambda\{(\alpha,\beta,\gamma,\lambda) \in o\}, \\ & \delta > -d\}. \end{aligned}$$

Let $\mathcal{O}^+$ (resp., $\mathcal{O}^-$) refer to the collection of the sets o^+ (resp., o^-). Note that $\mathcal{O}^+$ and $\mathcal{O}^-$ are collections of graphs of partially defined trivariate functions over

$\mathbb{L}_e$, and with a slight abuse of notation let $\mathcal{O}^+$ and $\mathcal{O}^-$ respectively denote the collections of these functions. (See Figure 1.) Define the *sandwich region* $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$ as

$$\Sigma(\mathcal{O}^-, \mathcal{O}^+) := \{(\alpha, \beta, \gamma, \delta) \in \mathcal{C}_e \mid \max_s s(\alpha, \beta, \gamma) \leq \delta \leq \min_t t(\alpha, \beta, \gamma), \quad s \in \mathcal{O}^-,\ t \in \mathcal{O}^+\}.$$

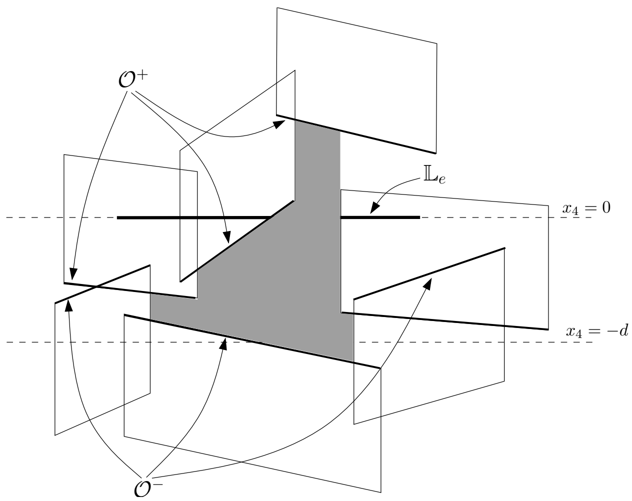

Figure 1: A schematic illustration of the notation used to define $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$. The $\mathcal{C}_e$-obstacles are shown as quadrilaterals in the ambient space $\mathcal{C}_e$. Note that the $\mathcal{C}_e$-obstacles all have height d and that the surfaces of $\mathcal{O}^+$ (resp., of $\mathcal{O}^-$) are pairwise disjoint. The domain $\mathbb{L}_e$ is shown as a thick line segment, and the sandwich region $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$ (shown shaded) is only defined over this domain.

Observation 3. *The set of vertices of $\mathcal{F}$ that correspond to free placements in which the interior of $\mathcal{L}$ touches e is isomorphic to a subset of the vertices of $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$.*

Proof: Consider a vertex v of $\mathcal{F}$ as described in the observation and let p denote the corresponding placement of $\mathcal{L}$. Since p touches e, v corresponds to some point $v_e = (\alpha, \beta, \gamma, \delta) \in \mathcal{C}_e$, such that $-d < \delta < 0$. Consider the vertical line $\ell = \{(\alpha, \beta, \gamma, \lambda) | \lambda \in \mathbb{R}\}$ spanned by v_e within $\mathcal{C}_e$. Since the placement p is free, the point v_e is not contained in $o \cap \ell$, for any $\mathcal{C}_e$-obstacle o. Note that the length of any such intersection $o \cap \ell$ is at least d. Thus any such intersection whose uppermost (resp., bottom) point lies below $(\alpha, \beta, \gamma, 0)$ (resp., above $(\alpha, \beta, \gamma, -d)$) has to lie below (resp., above) v_e. Therefore v_e is a point within the sandwich region $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$, and is thus a vertex of $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$. □

The complexity of $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$ is $O(n^3)$. Indeed, project the collections $\mathcal{O}^+$ and $\mathcal{O}^-$ onto $\mathbb{L}_e$ to obtain two collections of semi-algebraic sets of constant descriptive complexity in a three-dimensional space. Consider the arrangement $\mathcal{A}$ of the boundaries of these sets. Since the triangles of $\mathcal{S}$ are disjoint, the upper envelope of $\mathcal{O}^-$ is attained at any point by at most one surface $o^- \in \mathcal{O}^-$ and the same holds for the lower envelope of $\mathcal{O}^+$, which is attained at any point by at most one $o^+ \in \mathcal{O}^+$. Therefore, the overlay of the projections of these envelopes onto $\mathbb{L}_e$ is a substructure of $\mathcal{A}$ and has complexity $O(n^3)$. Over each cell of the overlay, the sandwich region has complexity $O(1)$, implying the bound $O(n^3)$ on the complexity of $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$.

Observations 2 and 3 suggest that we can bound the number of vertices of $\mathcal{F}$ by bounding the compounded complexity of the associated sandwich regions $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$ for all edges e. This implies the theorem. The upper bound is tight [22]. □

Remark: As a careful inspection of the proof can reveal, Theorem 3.1 still holds when $\mathcal{S}$ is a set of n pairwise disjoint semi-algebraic sets of constant descriptive complexity.

3.2 Cigars

Theorem 3.2. *The complexity of $\mathcal{F} = \mathcal{F}(\mathcal{M}, \mathcal{S})$ is $O(n^{4+\varepsilon})$.*

Proof: Consider the Minkowski sum $\tilde{\mathcal{S}} := \mathcal{S} \oplus \mathcal{B}$. It is easy to see that $\mathcal{F}(\mathcal{M}, \mathcal{S}) = \mathcal{F}(\mathcal{L}, \tilde{\mathcal{S}})$. As in the proof of Theorem 3.1, we can concentrate on counting the number of vertices of $\mathcal{F}$. Consider first the vertices that correspond to free placements of $\mathcal{L}$ in which its interior touches a feature $\tilde{e} := e \oplus \mathcal{B}$ of $\tilde{\mathcal{S}}$, where e is an edge of one of the triangles of $\mathcal{S}$. The number of these vertices can be bounded as follows. Fix such an edge e and consider the configuration space $\mathcal{C}_{\tilde{e}}$ of $\mathcal{L}$ with respect to $\tilde{\mathcal{S}}$ when $\mathcal{L}$ is tangent to $\tilde{e}$, defined similarly to the preceding section. For every $\mathcal{C}_{\tilde{e}}$-obstacle o, define o^+ and o^- as above. Let $\mathcal{O}^+$ (resp., $\mathcal{O}^-$) refer to the collection of the sets o^+ (resp., o^-) and let the sandwich region $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$ be defined as above. The following observation is proved in complete analogy to Observation 3.

Observation 4. *The set of vertices of $\mathcal{F}$ that correspond to free placements in which the interior of $\mathcal{L}$ touches $\tilde{e}$ are isomorphic to a subset of the vertices of $\Sigma(\mathcal{O}^-, \mathcal{O}^+)$.*

$\Sigma(\mathcal{O}^-, \mathcal{O}^+)$ is a sandwich region defined by $O(n)$ trivariate semi-algebraic functions of constant descriptive complexity. Its complexity is $O(n^{3+\varepsilon})$ [28]. Observation 4 thus implies that the number of vertices of $\mathcal{F}$ as above is $O(n^{4+\varepsilon})$.

Consider now the vertices that correspond to free placements of $\mathcal{L}$ in which its interior touches a feature $\tilde{f} := f \oplus \mathcal{B}$ of $\tilde{\mathcal{S}}$, where f is the interior face of one of the triangles of $\mathcal{S}$, but does not touch any feature $\tilde{e}$ as above. It is easy to see that such vertices correspond to vertices of the two-dimensional configuration space of $\mathcal{L}$ rigidly moving within a plane P_f parallel to f, among the obstacles $\tilde{s} \cap P_f$ for $\tilde{s} \in \tilde{\mathcal{S}}$. This space is a substructure of an arrangement in three dimensions, and its complexity is thus $O(n^3)$ [41]. (A sharper bound can be shown with a more careful analysis, but this is of no consequence to us here.) The overall number of vertices of $\mathcal{F}$ under consideration is therefore $O(n^4)$.

It remains to bound the number of vertices of $\mathcal{F}$ such that in the corresponding placements the interior of $\mathcal{L}$ is not incident to any feature of $\tilde{\mathcal{S}}$. The general position assumption implies that in a placement of $\mathcal{L}$ that corresponds to a vertex of $\mathcal{F}$, $\mathcal{L}$ is incident to five 2-faces of $\tilde{\mathcal{S}}$. Since its interior is not incident to features of $\tilde{\mathcal{S}}$ as above, at least one of the endpoints of $\mathcal{L}$ is incident to three or more 2-faces of $\tilde{\mathcal{S}}$, implying that it is incident to a vertex of $\tilde{\mathcal{S}}$. (In fact, the general position assumption implies that one endpoint of $\mathcal{L}$ is incident to three 2-faces of $\tilde{\mathcal{S}}$, while the other is incident to two such features.) The number of such vertices is $O(n^{2+\varepsilon})$ [6]. Fix such a vertex v. For a triangle $s \in \mathcal{S}$, consider the Minkowski sum $\tilde{s} := s \oplus \mathcal{B}$. Define a bivariate function whose domain is the unit sphere $\mathbb{S}^2$:

$$F_s(\gamma) := \min_u \| u - v \|, \qquad u \in \tilde{s},\ (u - v)/\| u - v \| = \gamma.$$

In other words, $F_s(\gamma)$ is the lower envelope of $\tilde{s}$ in the radial parametrization $\mathbb{S}^2 \times \mathbb{R}$ of $\mathbb{R}^3$ around v. Define in addition the function $H(\gamma) = d$ for $\gamma \in \mathbb{S}^2$. Define a lower envelope $\mathcal{E}$ as follows:

$$\mathcal{E}(\gamma) := \min_{f \in \{\mathcal{F}_s | s \in \mathcal{S}\} \cup \{H\}} f(\gamma).$$

Observation 5. *The set of vertices of $\mathcal{F}$ that correspond to free placements in which an endpoint of $\mathcal{L}$ touches v and the interior of $\mathcal{L}$ is not incident to any feature of $\tilde{\mathcal{S}}$ is isomorphic to a subset of the vertices of $\mathcal{E}$.*

Proof: Consider a vertex q of $\mathcal{F}$ as described in the observation and let p denote the corresponding placement of $\mathcal{L}$. Consider the endpoints v, u of $\mathcal{L}$ in the placement p, and the corresponding points $v' = (\alpha, \beta, 0)$ and $u' = (\alpha, \beta, d)$ in the radial parametrization $\mathbb{S}^2 \times \mathbb{R}$ of $\mathbb{R}^3$ around v. Since the line segment uv is disjoint in its interior from the obstacles $\tilde{\mathcal{S}}$, it follows that u' lies on or below the graphs of the functions $\mathcal{F}_s$ for all $s \in \mathcal{S}$. Since u lies on the boundaries of two obstacles $\tilde{s} \in \tilde{\mathcal{S}}$ and since by construction u' lies on the graph of the function H, it follows that u' is a vertex of $\mathcal{E}$. □

$\mathcal{E}$ is a lower envelope defined by $O(n)$ bivariate semi-algebraic functions of constant descriptive complexity. Its complexity is $O(n^{2+\varepsilon})$ [42]. The number of vertices of $\mathcal{F}$ that correspond to placements of $\mathcal{L}$ in which its interior is not incident to $\tilde{S}$ as above is thus $O(n^{4+\varepsilon})$. This completes the proof of the theorem. □

4 Motion Planning

4.1 Moving Needles

Theorem 4.1. *Given $\mathcal{L}$ and $\mathcal{S}$ as in Theorem 3.1, they can be preprocessed in time $O(n^{4+\varepsilon})$ into a data structure that, for two query placements α and β of $\mathcal{L}$, decides in time $O(\log n)$ whether a collision-free rigid motion of $\mathcal{L}$ between α and β exists and if so, can output such a motion in time asymptotically proportional to its complexity.*

Proof: Since by Theorem 3.1 the complexity of $\mathcal{F}(\mathcal{L}, \mathcal{S})$ is $O(n^4)$, a natural motion planning algorithm would be to preprocess $\mathcal{F}(\mathcal{L}, \mathcal{S})$ into a point location data structure by decomposing it into cells of constant descriptive complexity and constructing the decomposition using randomized divide-and-conquer. Given query placements α and β, we would locate the corresponding points in the data structure and determine reachability by searching the adjacency graph of the decomposition cells. However, since $\mathcal{F}(\mathcal{L}, \mathcal{S})$ is a substructure of an arrangement of surfaces in five dimensions, we lack a general output-sensitive decomposition scheme that would produce close to $O(n^4)$ cells. The vertical decomposition of $\mathcal{F}(\mathcal{L}, \mathcal{S})$ is known to have complexity $O(n^{6+\varepsilon})$ [27] and a refined analysis (which we omit) can yield a better bound of $O(n^{5+\varepsilon})$ due to structural properties of $\mathcal{F}(\mathcal{L}, \mathcal{S})$. This bound is still too high and it is thus necessary to devise an alternative approach.

Our solution fits within the category of motion planning algorithms that utilize the "retraction" approach [3, 9, 10, 11, 34, 35], as it constructs and searches within a graph $\mathcal{G}$. Due to the above reasoning, this graph does not correspond to a decomposition of $\mathcal{F}(\mathcal{L}, \mathcal{S})$ into cells of constant descriptive complexity and its construction requires some care. In what follows we begin by describing the preprocessing stage in which we construct $\mathcal{G}$ and a multi-level data structure that efficiently locates the vertex of $\mathcal{G}$ that corresponds to a given free placement of the robot. We then specify the query algorithm that first locates the vertices of $\mathcal{G}$ that correspond to α and β and then searches the graph for a path that connects these vertices (or establishes a lack thereof).

We conclude the proof with a verification of the algorithm's correctness. Space limitations necessitate omitting many details throughout this section.

The preprocessing stage—constructing the graph $\mathcal{G}$. Consider the set Γ of $3n$ lines spanned by the edges of the triangles of $\mathcal{S}$. Let $\hat{\mathcal{S}} := \mathcal{S} \cup \Gamma$. In what follows we regard Γ as "virtual obstacles" that facilitate the connectivity of $\mathcal{G}$.

Fix $e \in \Gamma$ and define $\mathbb{L}_e$, $\mathcal{C}_e$ as in the proof of Theorem 3.1. For every $\mathcal{C}_e$-obstacle o that corresponds to a set in $\hat{\mathcal{S}}$, define o^+ and o^- as in the proof of Theorem 3.1. Let $\mathcal{O}^+$ (resp., $\mathcal{O}^-$) refer to the collection of the sets o^+ (resp., o^-). Consider the *overlay* of the upper envelope of $\mathcal{O}^-$ and the lower envelope of $\mathcal{O}^+$ [28]. Project $\mathcal{O}^+$ and $\mathcal{O}^-$ onto the hyperplane $x_4 = 0$ to obtain two collections of semi-algebraic sets of constant descriptive complexity. Preprocess the arrangement $\mathcal{A}$ of the boundaries of these sets for point location by constructing its vertical decomposition $\mathcal{V}$ using randomized divide-and-conquer in time $O(n^{3+\varepsilon})$ [27, 42]. Note that $\mathcal{O}^+$ and $\mathcal{O}^-$ consist of pairwise disjoint surface patches. Thus the above overlay is a substructure of the arrangement $\mathcal{A}$. Since $\mathcal{V}$ is a decomposition of $\mathcal{A}$ into cells of constant descriptive complexity, it is also such a decomposition of the overlay. We now refine it in order to obtain a decomposition of the sandwich region $\Sigma_e = \Sigma(\mathcal{O}^-, \mathcal{O}^+)$.

Over each cell c of $\mathcal{V}$ the upper envelope of $\mathcal{O}^-$ is attained by at most one surface $o^- \in \mathcal{O}^-$ and the same holds for the lower envelope of $\mathcal{O}^+$, which is attained by at most one $o^+ \in \mathcal{O}^+$. We compute the intersection $o^- \cap o^+$ over the domain c, project it onto c and compute the vertical decomposition of this projection within c, in time $O(1)$. The refined decomposition of the overlay obtained in this manner induces a decomposition $\mathcal{D}_e$ of Σ_e. We construct a graph whose vertices are the cells of all dimensions of $\mathcal{D}_e$. A pair of vertices is connected by an edge if it corresponds to a pair of cells (of possibly different dimensions) the intersection of whose closures is not empty. For each cell c' of $\mathcal{D}_e$, we associate the vertex that corresponds to c' with an arbitrary point within c'. By construction, this point corresponds to a placement p of $\mathcal{L}$ that either touches e or, if $\mathcal{L}$ is translated continuously from p along the line spanned by p, $\mathcal{L}$ remains free throughout the movement until it touches e. (We call a placement of $\mathcal{L}$ as described a *placement that sees* e, as illustrated in Figure 2.)

Since the space $\mathcal{C}_e$ in which $\mathcal{D}_e$ is constructed is a subspace of $\mathcal{C}$, each cell c' of $\mathcal{D}_e$ is a set in $\mathcal{C}$. This set is incident to the boundary of the $\mathcal{C}$-obstacle that corresponds to e and may be incident to the boundary of $\mathcal{C}$-obstacles that correspond to other lines in Γ. Let $f \in \Gamma$ be one such line. Our construction ensures that

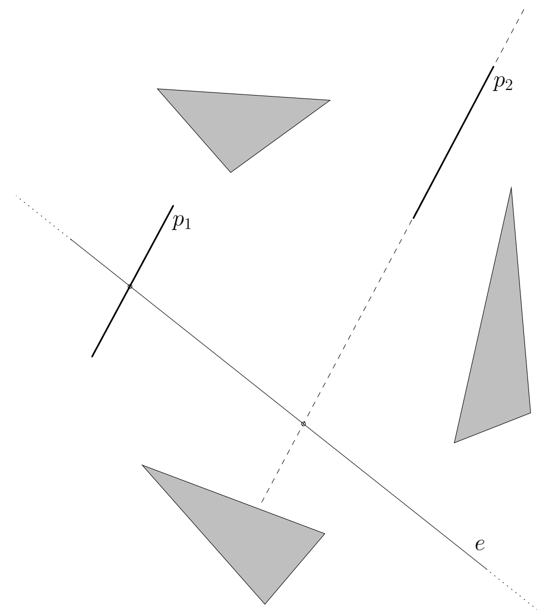

Figure 2: p_1 is a placement of $\mathcal{L}$ that touches e and p_2 is a placement that sees e.

c' is also a cell in the decomposition $\mathcal{D}_f$. We connect the two vertices of $\mathcal{G}$ that correspond to c' in $\mathcal{D}_e$ and $\mathcal{D}_f$, respectively, by an edge of $\mathcal{G}$. This matching of cells from different decompositions is done efficiently (in time $O(\log n)$ per edge) using the point location structures for each decomposition. This completes the construction of $\mathcal{G}$. We label each connected component of $\mathcal{G}$ with a unique identifying label (of length $O(\log n)$), which is stored at each vertex that belongs to the connected component.

The preprocessing stage—constructing auxiliary data structures. We also preprocess $\hat{\mathcal{S}}$ for certain types of queries that facilitate the identification of the vertex of $\mathcal{G}$ that corresponds to a query placement of $\mathcal{L}$. First, we preprocess $\hat{\mathcal{S}}$ into a data structure $\Upsilon(\hat{\mathcal{S}})$ that can answer ray shooting queries in time $O(\log n)$. We use any of the approaches of [14, 27, 37] to perform this construction in time $O(n^{4+\varepsilon})$.

We then preprocess $\hat{\mathcal{S}}$ into a data structure $\Phi(\hat{\mathcal{S}})$ that can answer the following type of *segment shooting* queries: Consider a query consisting of a line segment $s = (a, b) \subseteq \mathbb{R}^3$ and two parallel rays r_1 and r_2 originating at a and b respectively. Let Δ be the 1-parameter family of line segments parameterized by their distance from s, such that all segments of Δ are parallel to s and their endpoints lie on r_1 and r_2, respectively. Compute the minimal line segment s' of Δ that intersects $\hat{\mathcal{S}}$. See Figure 3 for an illustration. We sketch this data structure at a high level, leaving the details to the full version of the paper.

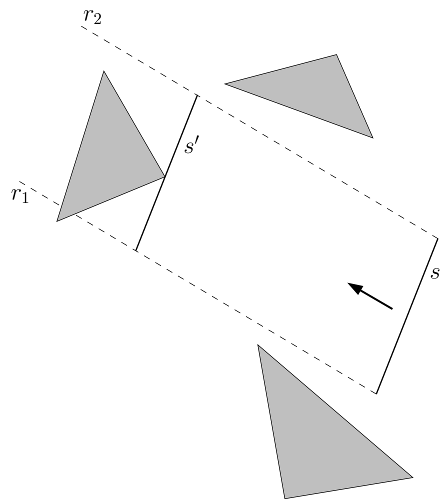

Figure 3: A segment shooting query and its answer.

We reduce the query to the following type of *quadrilateral emptiness* queries: Given a quadrilateral Q in $\mathbb{R}^3$, determine whether $Q \cap \hat{\mathcal{S}} = \emptyset$. (Often parametric searching [5, 33] is employed to reduce an optimization problem to a decision problem. As will be described in the full version of this paper, a more straightforward approach exists in our setting that does not entail a loss of asymptotic performance.) Use $\Upsilon(\hat{\mathcal{S}})$ to determine in time $O(\log n)$ whether the edges of Q intersect $\hat{\mathcal{S}}$. This reduces the query to *quadrilateral emptiness among lines*: Given a quadrilateral Q in $\mathbb{R}^3$, determine whether Q intersects any of the lines Γ.

We preprocess the lines Γ into a four-level range searching data structure $\Psi(\Gamma)$ (which is a component of $\Phi(\hat{\mathcal{S}})$) that answers triangular emptiness among lines in time $O(\log n)$. A line intersects Q if and only if it lies to a specific side (below or above) of each of the four lines spanned by the edges of Q. We transform the query lines into surfaces in the 4-dimensional Plücker space, and the lines spanned by the edges of Q into three points in this space. We preprocess the Plücker surfaces into a four-level range searching data structure, which allows us to determine whether any of these points has a specific above/below relation to the three query points. This completes the description of $\Psi(\Gamma)$. Using range searching machinery and the author's results on point location in four dimensions, $\Psi(\Gamma)$ can be constructed in time $O(n^{4+\varepsilon})$ such that a query can be answered in time $O(\log n)$, see [1, 27] and, e.g., [26] for a related simpler construction in three dimensions. This completes the description of the preprocessing stage. The following lemma is implied by the discussion above.

Lemma 4.1. *The graph $\mathcal{G}$ and the data structures $\Upsilon(\hat{\mathcal{S}})$, $\Phi(\hat{\mathcal{S}})$ can be constructed in time $O(n^{4+\varepsilon})$.*

The query algorithm. Given two query placements α and β, the query algorithm first uses $\Upsilon(\hat{\mathcal{S}})$ to determine whether α and β are free by shooting two rays along each placement, see Figure 4. Specifically, perform two ray shooting queries in $\Upsilon(\hat{\mathcal{S}})$, one directed from the bottom end-point of α (resp., β) towards the top one, and the other directed from the top end-point to the bottom one. Denote the points returned by the queries by f_α and c_α (resp., f_β and c_β) and note that they might lie at infinity. The placement is free if and only if both of these points lie outside of α (resp., β). We assume in what follows that this is the case, since otherwise the motion planning query has a trivial negative answer.

We now show how to construct a collision-free motion from α to a placement that corresponds to some vertex v_α of $\mathcal{G}$. It follows from the above that any free placement of $\mathcal{L}$ that sees e, for some $e \in \Gamma$, corresponds to a point in the sandwich region Σ_e in the configuration space $\mathcal{C}_e$. Moreover, since every cell c' of $\mathcal{D}_e$ is simply connected and has constant descriptive complexity, we can compute in time $O(1)$ a path from any point in c' to the point that is associated with the vertex of $\mathcal{G}$ that corresponds to c'. We thus only need to find a motion from α to a placement of $\mathcal{L}$ that sees some line e. This is accomplished as follows.

Denote the two triangles of $\mathcal{S}$ that contain the points f_α and c_α by f and c, respectively. (Intuitively, these are the triangles that are "seen" in both directions from α. We ignore the easier to handle case where one or both of f, c do not exist.) Consider the z-parallel plane Π in $\mathbb{R}^3$ spanned by α. Let ℓ_f and ℓ_c be the lines spanned by $f \cap \Pi$ and $c \cap \Pi$, respectively, and let ℓ_1 and ℓ_2 be lines parallel to ℓ_f and incident to f_α and c_α, respectively. Let $r_1 \subseteq \ell_1$ (resp., $r_2 \subseteq \ell_2$) be the ray that has the end-point $\ell_1 \cap \alpha$ (resp., $\ell_2 \cap \alpha$) and is disjoint from ℓ_c. Rays r_1 and r_2 satisfying these conditions necessarily exist.

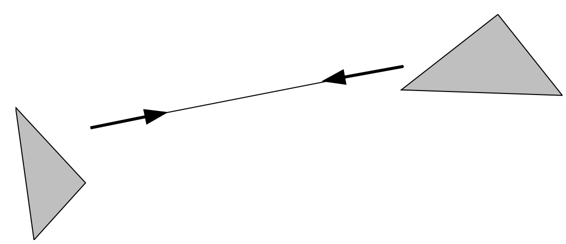

Figure 4: Using ray shooting to determine whether a placement is free.

Query $\Phi(\hat{\mathcal{S}})$ with α, r_1 and r_2. The data structure returns a line segment w of length d that is a free placement α' of $\mathcal{L}$, which corresponds to a point within some cell c' of $\mathcal{D}_e$. The shortest linear motion from

α to α' is collision-free by construction. As described above, a collision-free motion from α' to a placement that corresponds to a vertex v_α of $\mathcal{G}$ can be constructed in constant time.

Similarly, we construct a collision-free motion from β to a placement that corresponds to a vertex v_β of $\mathcal{G}$. If we are just interested in determining whether β is reachable from α, we simply check at this point whether v_β and v_α have a common label. The running time of the query algorithm up to this point is $O(\log n)$. If we are also interested in computing a collision-free motion from α to β, we find a path in $\mathcal{G}$ that connects v_α and v_β. Every vertex along the path corresponds to a free placement of $\mathcal{L}$. Every edge that connects two such vertices can be matched in time $O(1)$ to a collision-free motion between the corresponding placements. We combine the overall resulting collision-free motion with the motion from α to v_α in the beginning and the motion from v_β to β at the end. The resulting path is thus computed in time asymptotically proportional to its complexity. This completes the description of the query procedure. We summarize our findings in the following lemma.

LEMMA 4.2. *Given the graph $\mathcal{G}$ and the data structures $\Upsilon(\hat{\mathcal{S}})$ and $\Phi(\hat{\mathcal{S}})$, and given two query placements α and β of $\mathcal{L}$, a query procedure that determines the existence of a collision-free rigid motion of $\mathcal{L}$ between α and β runs in time $O(\log n)$. If it is determined that such a motion exists, the procedure can output such a motion in time asymptotically proportional to its complexity.*

Correctness. It is easy to see that whenever the query procedure asserts that there is a collision-free rigid motion of $\mathcal{L}$ from α to β, thus must be the case. (Indeed, it can be certified by the collision-free motion that can be produced by the algorithm.) However, proving that whenever a collision-free rigid motion of $\mathcal{L}$ from α to β exists, the query procedure correctly decides that α and β are inter-reachable is more involved. The main idea is as follows.

We show that any continuous collision-free motion Λ of $\mathcal{L}$ can be isomorphically mapped to a continuous collision-free motion Δ, such that any placement of $\mathcal{L}$ along Δ sees some line $e \in \Gamma$. Stated differently, the path in the configuration space $\mathcal{C}$ that corresponds to Δ is contained in the union of the sandwich regions Σ_e, for $e \in \Gamma$. The mapping is such that for any $\lambda \in \Lambda$ and $\delta \in \Delta$, such that λ is mapped to δ, there is a collision-free motion of constant complexity between λ and δ. Establishing the existence of such mapping forms the crux of the correctness proof. Omitting further details, we have:

LEMMA 4.3. *Given the graph $\mathcal{G}$ and the data structures $\Upsilon(\hat{\mathcal{S}})$ and $\Phi(\hat{\mathcal{S}})$, and given two query placements α and β of $\mathcal{L}$, the query procedure described above correctly determines whether a collision-free rigid motion of $\mathcal{L}$ between α and β exists.*

This completes the proof of Theorem 4.1. □

4.2 Moving Cigars Due to space limitations in this extended abstract we do not elaborate on the motion planning algorithm for cigar-shaped robots and only state the main result.

THEOREM 4.2. *Given $\mathcal{M}$ and $\mathcal{S}$ as in Theorem 3.2, they can be preprocessed in time $O(n^{4+\varepsilon})$ into a data structure that, for two query placements α and β of $\mathcal{M}$, decides in time $O(\text{polylog}\, n)$ whether a collision-free rigid motion of $\mathcal{M}$ between α and β exists and if so, can output such a motion in time asymptotically proportional to its complexity.*

5 Extremal Placement

Let $\mathcal{S}$ be be a collection of n closed pairwise interior-disjoint triangles. Let $\mathcal{U} \subseteq \mathbb{R}^3 \setminus \mathcal{S}$ denote the union of bounded connected components of $\mathbb{R}^3 \setminus \mathcal{S}$. Note that in this section the triangles of $\mathcal{S}$ have to be in degenerate position to make the problem non-trivial.

THEOREM 5.1. *The longest line segment that can be placed in the interior of $\mathcal{U}$ can be found in time $O(n^{4+\varepsilon})$.*

Proof: We claim that there is a longest line segment that touches an edge of a triangle of $\mathcal{S}$. Indeed, assume the contrary and consider the longest line segment l, which is not incident to an edge of $\mathcal{S}$. The endpoints of l are necessarily incident to triangles of $\mathcal{S}$. Denote these triangles by s and t. Consider sliding l parallel to the x-axis, while keeping its end-points incident to s and t, respectively. Due to the linearity of s and t, when l slides either in the positive or the negative x-direction as above, the length of l does not decrease. We perform this sliding until one of the endpoints of l reaches an edge of s or t, or the interior of l hits an edge of another triangle of $\mathcal{S}$. One of these is bound to happen. This produces a line segment that is not shorter than l that is incident to an edge of $\mathcal{S}$, a contradiction.

Let $\mathcal{L}$ now be an infinitesimally short line segment. Fix an edge e of $\mathcal{S}$ that is contained in the closure of $\mathcal{U}$ and construct the decomposition $\mathcal{D}_e$ of the sandwich region Σ_e, defined with respect to $\mathcal{L}$ and $\mathcal{S}$ as in the proof of Theorem 4.1. Each vertical segment in Σ_e corresponds to a free line segment that is tangent to e in $\mathbb{R}^3$ and vice versa. Due to the above observation it is thus sufficient to find the maximal vertical segment of Σ_e, for all e. Since each cell of $\mathcal{D}_e$ has constant descriptive complexity, the maximal vertical segment in each cell can be located in constant time. The maximal segment tangent to e can thus be located in time proportional to the complexity of $\mathcal{D}_e$, which is $O(n^3)$. This completes the proof of the theorem. □

THEOREM 5.2. *Let $\mathcal{M}$ be a cigar. The maximal similar copy of $\mathcal{M}$ that can be placed in the interior of $\mathcal{U}$ can be found in time $O(n^{4+\varepsilon})$.*

Proof: The algorithm relies on parametric searching [33], which reduces the problem to the following decision problem: Given a cigar $\mathcal{M}'$ *(which is a similar copy of* $\mathcal{M}$*)*, decide whether $\mathcal{M}'$ can be placed in the interior of $\mathcal{U}$. Parametric searching requires the derivation of an efficient parallel algorithm for this decision problem. We have devised such an algorithm that runs in time $O(\log n)$ and uses $O(n^{4+\varepsilon})$ processors. All further details are omitted from this extended abstract. □

6 Discussion and Open Problems

In this paper we have provided near-optimal algorithms for planning the motion of needles and cigars in three dimensions, almost matching the existing lower bound [22] and improving the best previous upper bounds by at least one order of magnitude [10, 22]. These are the first near-optimal rigid-motion planning algorithms in three dimensions that utilize the specific geometric structure of the problem.

There is a large variety of natural robot motion planning problems in three dimensions that can be considered following our work. Other types of robots with five degrees of freedom in three dimensions include circles, cylinders and general bodies of revolution. We conjecture that rigid-motion planning algorithms with near-$O(n^4)$ running time exist for these types of robots as well. A more challenging undertaking is to consider robots with six degrees of freedom in 3-space, such as triangles, boxes and general polyhedra. Do near-$O(n^4)$ motion planning algorithms exist for such shapes, or are there super-$\Omega(n^4)$ lower bounds? We are leaving this question to future research.

Acknowledgements

I wish to thank Micha Sharir for bringing the needle motion planning problem to my attention and for subsequent discussions, Danny Halperin for his encouragement, Mark Overmars for suggesting to consider cigar-shaped robots, and Eti Ezra, Jeff Erickson, Steve LaValle and Sariel Har-Peled for helpful discussions.

References

[1] P. K. Agarwal. Range searching. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry, Second Edition*, pages 809–837. CRC Press LLC, Boca Raton, FL, 2004.

[2] P. K. Agarwal, N. Amenta, and M. Sharir. Largest placement of one convex polygon inside another. *Discrete Comput. Geom.*, 19:95–104, 1998.

[3] P. K. Agarwal, B. Aronov, and M. Sharir. Motion planning for a convex polygon in a polygonal environment. *Discrete Comput. Geom.*, 22:201–221, 1999.

[4] P. K. Agarwal, E. Flato, and D. Halperin. Polygon decomposition for efficient construction of minkowski sums. *Computational Geometry: Theory and Applications*, 21:39–61, 2002.

[5] P. K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM Journal on Computing*, 22(4):794–806, 1993.

[6] P. K. Agarwal and M. Sharir. Pipes, cigars, and kreplach: The union of Minkowski sums in three dimensions. *Discrete Comput. Geom.*, 24(4):645–685, 2000.

[7] B. Aronov and M. Sharir. On translational motion planning of a convex polyhedron in 3-space. *SIAM J. Comput.*, 26:1785–1803, 1997.

[8] F. Avnaim, J.-D. Boissonnat, and B. Faverjon. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. volume 391 of *Lecture Notes Comput. Sci.*, pages 67–86. 1989.

[9] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets. In *Proc. 28th Annu. ACM Sympos. Theory Comput.*, pages 168–173, 1996.

[10] J. Canny. *The Complexity of Robot Motion Planning.* ACM – MIT Press Doctoral Dissertation Award Series. MIT Press, Cambridge, MA, 1988.

[11] J. Canny. Computing roadmaps in general semialgebraic sets. *Comput. J.*, 36:409–418, 1993.

[12] B. Chazelle. The polygon containment problem. In F. P. Preparata, editor, *Computational Geometry*, volume 1 of *Adv. Comput. Res.*, pages 1–33. JAI Press, Greenwich, Conn., 1983.

[13] L. P. Chew and K. Kedem. A convex polygon among polygonal obstacles: Placement and high-clearance motion. *Comput. Geom. Theory Appl.*, 3:59–89, 1993.

[14] M. de Berg, D. Halperin, M. Overmars, J. Snoeyink, and M. van Kreveld. Efficient ray shooting and hidden surface removal. *Algorithmica*, 12:30–53, 1994.

[15] L. J. Guibas, M. Sharir, and S. Sifrony. On the general motion planning problem with two degrees of freedom. *Discrete Comput. Geom.*, 4:491–521, 1989.

[16] D. Halperin. Personal communication.

[17] D. Halperin, L. E. Kavraki, and J.-C. Latombe. Robotics. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry, Second Edition*, chapter 48, pages 1065–1093. CRC Press LLC, Boca Raton, FL, 2004.

[18] D. Halperin, M. H. Overmars, and M. Sharir. Efficient

motion planning for an L-shaped object. *SIAM J. Comput.*, 21:1–23, 1992.

[19] D. Halperin and M. Sharir. A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. *Discrete Comput. Geom.*, 16:121–134, 1996.

[20] D. Halperin and C.-K. Yap. Combinatorial complexity of translating a box in polyhedral 3-space. *Comput. Geom. Theory Appl.*, 9:181–196, 1998.

[21] S. Har-Peled, T. M. Chan, B. Aronov, D. Halperin, and J. Snoeyink. The complexity of a single face of a Minkowski sum. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 91–96, 1995.

[22] Y. Ke and J. O'Rourke. Moving a ladder in three dimensions: upper and lower bounds. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 136–145, 1987.

[23] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.

[24] K. Kedem and M. Sharir. An efficient motion planning algorithm for a convex rigid polygonal object in 2-dimensional polygonal space. *Discrete Comput. Geom.*, 5:43–75, 1990.

[25] K. Kedem, M. Sharir, and S. Toledo. On critical orientations in the Kedem-Sharir motion planning algorithm for a convex polygon in the plane. *Discrete Comput. Geom.*, 17:227–240, 1997.

[26] V. Koltun. Segment intersection searching problems in general settings. *Discrete and Computational Geometry*, 30:25–44, 2003.

[27] V. Koltun. Almost tight upper bounds for vertical decompositions in four dimensions. *Journal of the ACM*, to appear.

[28] V. Koltun and M. Sharir. The partition technique for overlays of envelopes. *SIAM Journal on Computing*, 32:841–863, 2003.

[29] D. Leven and M. Sharir. An efficient and simple motion planning algorithm for a ladder moving in two-dimensional space amidst polygonal barriers. *J. Algorithms*, 8:192–215, 1987.

[30] D. Leven and M. Sharir. On the number of critical free contacts of a convex polygonal object moving in two-dimensional polygonal space. *Discrete Comput. Geom.*, 2:255–270, 1987.

[31] D. Leven and M. Sharir. Planning a purely translational motion for a convex object in two-dimensional space using generalized Voronoi diagrams. *Discrete Comput. Geom.*, 2:9–31, 1987.

[32] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570, 1979.

[33] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30(4):852–865, 1983.

[34] C. Ó'Dúnlaing, M. Sharir, and C. K. Yap. Generalized Voronoi diagrams for a ladder: I. Topological analysis. *Commun. Pure Appl. Math.*, 39:423–483, 1986.

[35] C. Ó'Dúnlaing, M. Sharir, and C. K. Yap. Generalized Voronoi diagrams for a ladder: II. Efficient construction of the diagram. *Algorithmica*, 2:27–59, 1987.

[36] J. O'Rourke. A lower bound on moving a ladder. Report JHU/EECS-85/20, Dept. Comput. Sci., Johns Hopkins Univ., Baltimore, MD, 1985.

[37] M. Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9:471–494, 1993.

[38] M. Pellegrini. Ray shooting and lines in space. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry, Second Edition*, pages 839–856. CRC Press LLC, Boca Raton, FL, 2004.

[39] J. T. Schwartz and M. Sharir. On the "piano movers" problem II: General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[40] J. T. Schwartz and M. Sharir. On the "piano movers" problem V: the case of a rod moving in three-dimensional space amidst polyhedral obstacles. *Commun. Pure Appl. Math.*, 37:815–848, 1984.

[41] M. Sharir. Algorithmic motion planning. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry, Second Edition*, pages 1037–1064. CRC Press LLC, Boca Raton, FL, 2004.

[42] M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, New York, 1995.

[43] M. Sharir and S. Toledo. Extremal polygon containment problems. *Comput. Geom. Theory Appl.*, 4:99–118, 1994.

[44] S. Sifrony and M. Sharir. A new efficient motion-planning algorithm for a rod in two-dimensional polygonal space. *Algorithmica*, 2:367–402, 1987.

[45] A. F. van der Stappen and M. H. Overmars. Motion planning amidst fat obstacles. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 31–40, 1994.

[46] G. Vegter. The visibility diagram: A data structure for visibility problems and motion planning. In *Proc. 2nd Scand. Workshop Algorithm Theory*, volume 447 of *Lecture Notes Comput. Sci.*, pages 97–110. Springer-Verlag, 1990.