Resolution-Optimal Motion Planning for Steerable Needles

Mengyu Fu[1], Kiril Solovey[2], Oren Salzman[2], and Ron Alterovitz[1]

***Abstract*— Medical steerable needles can follow 3D curvilinear trajectories inside body tissue, enabling them to move around critical anatomical structures and precisely reach clinically significant targets in a minimally invasive way. Automating needle steering, with motion planning as a key component, has the potential to maximize the accuracy, precision, speed, and safety of steerable needle procedures. In this paper, we introduce the first resolution-optimal motion planner for steerable needles that offers excellent practical performance in terms of runtime while simultaneously providing strong theoretical guarantees on completeness and the global optimality of the motion plan in finite time. Compared to state-of-the-art steerable needle motion planners, simulation experiments on realistic scenarios of lung biopsy demonstrate that our proposed planner is faster in generating higher-quality plans while incorporating clinically relevant cost functions. This indicates that the theoretical guarantees of the proposed planner have a practical impact on the motion plan quality, which is valuable for computing motion plans that minimize patient trauma.**

I. Introduction

Medical steerable needles have the potential to improve patient care in diagnostic and therapeutic procedures including biopsy, localized drug delivery, and radioactive seed implantation for cancer treatment [2]. Steerable needles have a small diameter and are made of a highly flexible material, which allows them to follow 3D curvilinear trajectories inside the tissue. These properties enable steerable needles to move around critical anatomical structures to reduce patient trauma and reach sites previously unreachable with traditional straight needles [3–6].

Automating needle steering can improve the accuracy, precision, speed, and safety of steerable needle procedures. Automating these procedures can also facilitate their broad use, since manual control of a steerable needle is challenging due to the nonholonomic constraints on the needle's motion and the high level of precision required to operate it. A key component of automating steerable needle procedures is motion planning: computing feasible, obstacle-avoiding trajectories through the tissue to reach a target. The trajectory of the needle through tissue should also maximize patient safety, which can be quantified using metrics such as minimizing trajectory length [7], maximizing clearance from obstacles [8–11], and minimizing damage to sensitive tissue [12]. An example is shown in Fig. 1.

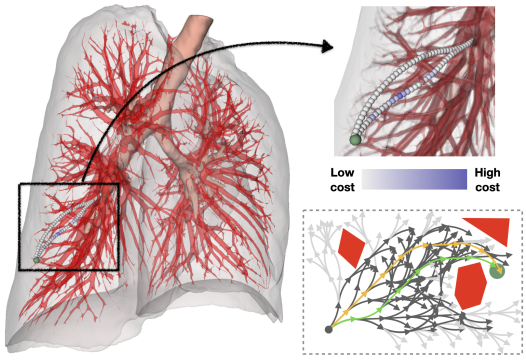


Fig. 1: **Left:** Overview of two different steerable-needle motion plans, both reaching a nodule (green) in the lung parenchyma for biopsy or cancer treatment while avoiding critical anatomical structures such as the bronchial tubes (brown) and major blood vessels (red). **Top right:** A zoomed-in view of two different plans where small blood vessels are rendered in grayscale. We use the method in [12] to reconstruct a cost map that represents the risk of puncturing small blood vessels. Different colors along the plans show different costs on the cost map. The top plan is computed with our proposed planner, RCS*. The bottom plan is computed with our previous RCS algorithm [30] and has a higher cost. **Bottom right:** A 2D illustration of the tree grown using our resolution-optimal motion planner towards a goal region (green) while avoiding obstacles (red). The best (shortest) plan found is shown in light green, and another valid but worse (longer) plan is shown in yellow. With generic and domain-specific optimizations, we can greatly shrink the search space (i.e., discard the light gray edges).

A motion planner for steerable needles should ideally offer guarantees on both finite-time completeness (i.e., compute in finite time a motion plan or indicate that none exists) and optimality (i.e., return a globally optimal motion plan with respect to a chosen cost metric). Most prior motion planners for steerable needles lack one or both of these criteria. For instance, some methods for steerable needle motion planning lack completeness guarantees [7, 13–18], and so may fail to find a motion plan when one exists. Some methods do aim to optimize motion plan cost but they lack *global* optimality guarantees [7, 18, 19].

Some sampling-based planners are known to be both complete and optimal, albeit those properties are usually proven only for an asymptotic regime where the number of samples tends to infinity [20–27]. Thus, it is unclear what should be the number of samples necessary to achieve those guarantees in practice. Recent work has developed optimality guarantees for finite sampling, although those results cannot be currently applied to steerable needles as they deal with holonomic systems [28, 29].

In this paper, we introduce the first motion planner for steerable needles that offers excellent practical performance in terms of runtime while providing strong theoretical guarantees on completeness and the cost of the motion plan in *finite time*. In particular, we consider a specific type of optimality in relation to the motion plan cost—resolution

This project was supported in part by the United States National Institutes of Health (NIH) under award R01EB024864; the United States National Science Foundation (NSF) under awards 2008475 and 2038855; the Israeli Ministry of Science, Technology and Space (MOST) under awards 3-17385 and 3-16079; and the United States-Israel Binational Science Foundation (BSF) under award 2019703.

Code is available at [1].

[1]M. Fu and R. Alterovitz are with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA. {mfu,ron}@cs.unc.edu

[2]K. Solovey and O. Salzman are with Computer Science Department, Technion - Israel Institute of Technology, Israel. kirilsol@stanford.edu, osalzman@cs.technion.ac.il

optimality. Generally speaking, a resolution characterizes the discretization of some space (e.g., state space, configuration space, action space, and time). An algorithm is *resolution complete* if there exists a fine-enough resolution with which the algorithm finds a motion plan in finite time when a qualified motion plan exists, and otherwise correctly returns that no such plan exists [30]. An algorithm is *resolution optimal* if it is resolution complete and if, when it does return a motion plan, the plan's cost is guaranteed to be within a desired approximation factor of the cost of a globally optimal qualified motion plan.

Our new motion planner builds on Resolution-Complete Search (RCS) [30], which is a resolution-complete but not resolution-optimal motion planner for steerable needles. If a motion plan exists, RCS would find a motion plan in finite time assuming that the parameter resolution is fine enough, but it provides no guarantees on the motion plan cost. To achieve resolution optimality, we enhance RCS with cost-aware duplicate pruning while incorporating motion plan cost tracking and a heuristic function to improve efficiency. We provide a proof sketch to show the resolution optimality of our method with a careful discussion of assumptions and required conditions. We also demonstrate experimentally on a realistic lung biopsy scenario that our new method, RCS*, outperforms the state-of-the-art in terms of runtime and plan quality.

II. Related Work

A. Motion planning for steerable needles

A variety of approaches have been proposed for the motion planning of steerable needles. Duindam et al. [13] proposed a planner based on inverse kinematics but provided no theoretical guarantees. Liu et al. [18] developed the Adaptive Fractal Tree (AFT) for needle steering. Their method iteratively refines the lowest-cost plan from the previous iteration, but refining the best plan of a coarse resolution does not necessarily lead to the best plan in a finer resolution. Pinzi et al. [19] extended it to account for goal orientation constraints.

Some planners adapt sampling-based methods such as Rapidly-exploring Random Tree (RRT) [20] for steerable needles. Xu et al. [31] used an RRT variant for needle steering but showed low time efficiency. Patil et al. [15] developed an RRT-based needle planner that samples in the 3D workspace rather than the configuration space. Sampling in a lower-dimension space and their customized distance function made the planner work efficiently in many practical cases, but this also invalidates the probabilistic-completeness guarantee of RRT [20, 23]. To avoid dealing with curvature constraints directly, Favaro et al. [7] proposed a hybrid method to combine sampling and smoothing. First, a tree embedded in the 3D workspace is built with RRT* [24], then candidate plans found by the tree are smoothed to further account for the curvature constraint. However, such a decoupling invalidates the asymptotic optimality guarantee [24, 25]. Sun et al. [27] proposed a needle planner by building multiple RRTs, which is asymptotically optimal when the number of trees tends to infinity. Other methods focus on accounting for the uncertainty during needle insertion without providing formal guarantees [14, 16, 17].

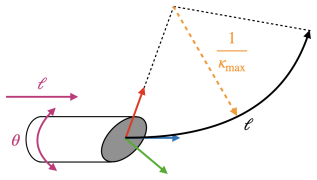
B. Resolution-optimal motion planners

Although resolution completeness has been frequently mentioned [32–36], resolution optimality earned little attention, possibly due to being rather complex to analyze mathematically, particularly for nonholonomic systems. Consequently, many planners developed for nonholonomic systems focus on asymptotic optimality instead [22, 26, 37, 38].

Barraquand et al. [33] proposed a resolution-complete planner for single or multiple robots with nonholonomic constraints. Their method is also optimal with respect to the number of reverse maneuvers in the plan. Pivtoraiko et al. [39] proposed the idea of motion planning using state lattices for field robots. Their state lattices planner is resolution optimal since the search is optimal for a graph of some resolution and the discrete state grid approximates the continuous space as resolution increases. Ljungqvist et al. [40] later extended [39] for a general two-trailer system in 2D. However, these methods are designed for large-scale workspaces, making them unsuitable for tasks where a high level of precision is required, such as for steerable needles.

III. Problem Definition

We consider bevel-tip flexible steerable needles [3–6] controlled by insertion and rotation at their base. A bevel-tip steerable needle is made of a highly flexible material such that, when being inserted through tissue, asymmetric forces applied by the bevel cause the needle to take a curved trajectory. The maximum curvature of the needle's path in the tissue, $\kappa_{\max}$, is influenced by the mechanical design of the needle and the tissue that the needle moves through. Additionally, axially rotating the needle at its base changes the direction the bevel is facing, enabling us to control the steering direction. The figure to the right illustrates the needle's kinematics, where the needle can be inserted by length ℓ and axially rotated at its base by θ.



We make the common assumption that the steerable needle is sufficiently flexible so the needle shaft moves along the trajectory created by its tip while the lateral motions are negligible. Thus, the configuration space of the steerable needle is defined by the pose of its tip, $\mathcal{X} \subset \mathcal{SE}(3)$, where a given configuration $\mathbf{x} = (p, q) \in \mathcal{X}$, specifies the position and orientation components $p \in \mathbb{R}^3$ and $q \in \mathcal{SO}(3)$, respectively. We assume that $\mathcal{X}$ is compact. We denote the 3D workspace by $\mathcal{W} \in \mathbb{R}^3$, a subset of which is occupied by obstacles $\mathcal{W}_{\text{obs}} \subset \mathcal{W}$. A configuration $\mathbf{x} = (p, q)$ is *collision free* if and only if $p \notin \mathcal{W}_{\text{obs}}$. We define $\mathcal{X}_{\text{free}}$ as the union of all collision-free configurations.

A motion plan of the needle is a trajectory $\sigma : [0, \ell] \to \mathcal{X}$, where ℓ is the length of the trajectory. We also use the notation ℓ_σ to denote the length of σ. A motion plan (or trajectory) σ is *collision free* if $\forall s \in [0, \ell], \sigma(s) \in \mathcal{X}_{\text{free}}$. To evaluate the quality of a motion plan, we consider a configuration-based cost function $c : \mathcal{X} \to \mathbb{R}$. We require c to be well behaved (formally defined in Sec V), which includes being Lipshitz continuous and bounded within $[c_{\min}, c_{\max}]$.

We define the cost of a motion plan as the integral of the configuration-based cost along a given trajectory σ, i.e., $\mathcal{C}(\sigma) = \int_0^\ell c(\sigma(s))\mathrm{d}s$. This definition captures a variety of cost functions, including trajectory length and integrating over a cost map derived from medical images.

A motion plan is (kinematically) *feasible* if the curvatures along the trajectory never exceed $\kappa_{\max}$. A motion plan is *valid* if it is collision free and feasible. We are now ready to state the steerable needle motion planning problem.

Problem 1. An optimal steerable needle motion planning problem is defined as the tuple $\Delta = (\mathcal{X}, \mathcal{W}_{\text{obs}}, \mathbf{x}_{\text{start}}, p_{\text{goal}}, \tau, \ell_{\max}, \kappa_{\max}, \mathcal{C})$, where $\mathcal{W}_{\text{obs}}$ is the obstacle set, $\mathbf{x}_{\text{start}}$ is the start configuration, $p_{\text{goal}} \in \mathcal{W}$ is the goal point, $\tau > 0$ is the goal tolerance, $\ell_{\max}$ is the maximum insertion length, $\kappa_{\max}$ is the maximum curvature, and $\mathcal{C}$ is a cost function. The problem calls for computing an optimal valid motion plan $\sigma^* = \text{argmin}_\sigma \mathcal{C}(\sigma)$ subject to:

$$\begin{aligned}&\sigma \text{ is valid},\\ &\sigma(0) = \mathbf{x}_{\text{start}},\\ &\ell_\sigma \le \ell_{\max},\\ &\|\text{Proj}(\sigma(\ell_\sigma)) - p_{\text{goal}}\|_2 \le \tau,\end{aligned}$$

where $\text{Proj}(\mathbf{x}) = p$ for $\mathbf{x} = (p, q)$.

As we show in Sec. V, for any given instance of Problem 1, under some mild assumptions, there exists a fine-enough cutoff resolution $R_{\min} = \{\delta\ell_{\min}, \delta\theta_{\min}\}$ (corresponding to the needle's insertion and axial rotation, respectively) for which our planner is guaranteed to return a motion plan with a cost to be within a desired approximation factor of a globally optimal qualified motion plan in finite time (if any qualified motion plan exists), or indicate that no qualified motion plan exists. Similar to [30], we assume there exist minimal motions that are precisely achievable by the hardware system in tissues, informing the cuttoff resolution. In our specific case, the minimal motions are the minimal insertion and minimal rotation the needle tip can precisely perform, determined by a lower-level needle controller.

IV. The RCS* algorithm

We describe our RCS* algorithm for resolution-optimal motion planning. We also highlight the differences between our new algorithm and our previous method RCS [30]. After presenting RCS*, we discuss an additional procedure to further improve its performance.

A. Algorithm description

The core idea of RCS* (and RCS) is to build a search tree with predefined motions of multiple resolutions. Specifically, RCS* constructs a search tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ embedded in the configuration space, where each node $v \in \mathcal{V}$ is associated with a configuration $\mathbf{x}_v \in \mathcal{X}$ and each edge $e = (u, v) \in \mathcal{E}$ represents a transition from $\mathbf{x}_u$ to $\mathbf{x}_v$. The search tree is rooted at the given start configuration $\mathbf{x}_{\text{start}}$ and explores valid motion plans when expanded in the configuration space. See the pseudocode of RCS* in Alg. 1.

To generate new nodes, RCS* expands existing nodes with predefined (kinematically) feasible motion primitives [41]. In RCS*, a motion primitive defines a local circular trajectory with some constant curvature $\kappa \le \kappa_{\max}$, some length $\delta\ell > 0$, and some curving direction $\delta\theta \in [0, 2\pi)$. That is, a motion primitive is defined as a tuple $\mathcal{M} = (\kappa, \delta\ell, \delta\theta)$. We denote the operation of applying a motion primitive $\mathcal{M}$ to configuration $\mathbf{x}_v$ as $\mathbf{x}_u = \mathbf{x}_v \oplus \mathcal{M}$, where $\mathbf{x}_u$ is the resultant configuration. RCS* uses a fixed set of curvatures $\{0, \kappa_{\max}\}$ and defines the resolution of a motion primitive as a function of $\delta\ell$ and $\delta\theta$, since any curvature $\kappa \in [0, \kappa_{\max}]$ can be well approximated by interleaving curvature 0 and $\kappa_{\max}$ [30]. Generally speaking, the finer a resolution is, the finer the intervals $[0, \delta\ell_{\max}]$ and $[0, \delta\theta_{\max}]$ are discretized. We mention that the coarsest resolution is set with a user-defined $\delta\ell_{\max}$ and $\delta\theta_{\max} = \pi/2$ (4 initial steering orientations as shown in line 1).

Algorithm 1 RCS*
Input: $\mathcal{W}_{\text{obs}}, \mathbf{x}_{\text{start}}, p_{\text{goal}}, \tau, \kappa_{\max}, \ell_{\max}, \delta\ell_{\max}, R_{\min}$

1: $\Theta \leftarrow \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}, K \leftarrow \{0, \kappa_{\max}\}$
2: root $\leftarrow (\mathbf{x}_{\text{start}}, 0, 0)$ ▷ The root has rank 0 and cost 0
3: OPEN $\leftarrow \{\text{root}\}$, CLOSED $\leftarrow \emptyset$, bestPlan $\leftarrow$ NULL
4: **while** not OPEN.empty() **do**
5: $v \leftarrow$ OPEN.extract()
6: **if** Valid$(v, \mathcal{W}_{\text{obs}}, p_{\text{goal}}, \ell_{\max})$ **then**
7: **if not** CLOSED.existDuplicate(v) **then**
8: **if** GoalReached$(v, p_{\text{goal}}, \tau)$ **then**
9: bestPlan.update(v)
10: **for** $\mathcal{M} \in$ Primitives$(K, \delta\ell_{\max}, \Theta)$ **do**
11: OPEN.insert$(v \oplus \mathcal{M})$
12: CLOSED.insert(v)
13: **if** v != root **then**
14: **for** $\mathcal{M} \in$ RefinedPrimitives$(\mathcal{M}_v)$ **do**
15: **if** ValidResolution$(\mathcal{M}, R_{\min})$ **then**
16: OPEN.insert$(v.\text{parent} \oplus \mathcal{M})$
17: **return** bestPlan

In each iteration of RCS*, an expansion of existing nodes is performed in an A* fashion. In particular, nodes are iteratively extracted from the OPEN list (line 5), wherein nodes are ordered according to their *rank* and a secondary metric $f(\cdot)$. We define the rank of a node as a function of the node's depth in the tree and the resolution of the motion primitives leading to the node. The deeper a node is in the tree and the finer resolution the motion primitives are, the higher rank a node has (see formal definition of rank in [30]). The secondary metric $f(v) = \mathcal{C}(v) + h(v)$ has $\mathcal{C}(v)$ denoting the cost of the trajectory from the root of $\mathcal{T}$ to v with respect to $\mathcal{C}$ and $h(v)$ being a heuristic function estimating the cost of the trajectory from the node v to the goal point. For example, in the case where $\mathcal{C}$ is trajectory length, we have $h(v)$ be the length of the Dubins curve [42] on the plane spanned by $\mathbf{x}_v$ and p_{goal}. Unlike in RCS, where nodes with lower rank are always extracted first, RCS* relaxes this ordering by introducing a *look-ahead* parameter denoted as $n_{\text{la}} \in \mathbb{N}$ (similar to the idea in [35]). At any time during the search, we denote the minimum rank of nodes in the OPEN list as r_{open}. Then we order all nodes with rank $r \le r_{\text{open}} + n_{\text{la}}$ according to a secondary metric $f(\cdot)$. This is done to prioritize searching nodes from a coarser resolution, which speeds up finding an initial motion plan.

Given an extracted node v, we first check if it is valid (line 6) using the conditions described for RCS, which

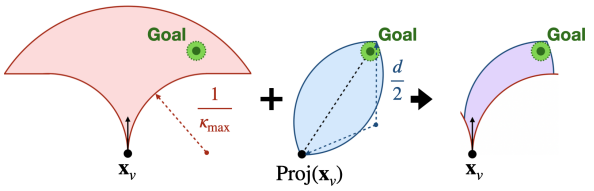


Fig. 2: A 2D illustration of the approximated reachable workspace. The kinematically forward-reachable workspace is shaded in red. The feasible workspace is shaded in blue. The diameter of the circular arcs is $d = \max(2/\kappa_{\max}, \tau + \|\mathrm{Proj}(\mathbf{x}_v) - p_{\mathrm{goal}}\|_2)$. The final approximated reachable workspace is shaded in purple.

ensure that (i) the insertion length would not exceed $\ell_{\max}$, (ii) the goal region is still reachable after getting to v, (iii) the trajectory from the root to v is not identical to another node that only needs coarser motion primitives to get to, and (iv) that the edge leading to v is collision free [30]. In addition, RCS* checks that the cost $\mathcal{C}(v)$ is smaller than the cost of the best plan reaching the goal region found so far. If the heuristic function $h(\cdot)$ is admissible, we use $f(v)$ instead of $\mathcal{C}(v)$ in the last condition, as f provides a better estimate of the node cost and hence allows to prune more vertices.

To further boost efficiency, the algorithm avoids expanding nodes that are overly similar to existing nodes in terms of the induced configuration and cost by performing duplicate detection (line 7). A node v is determined as a duplicate if there exists a node u in the CLOSED list that satisfies (i') $\rho(\mathbf{x}_u, \mathbf{x}_v) < d_{\mathrm{sim}}$ and (ii') $\mathcal{C}(u) \leq \mathcal{C}(v)$, where ρ is a distance function defined on $\mathcal{X}$ and d_{sim} is a similarity parameter. We use $\rho(\mathbf{x}_u, \mathbf{x}_v) = \|p_u - p_v\|_2 + \alpha \cdot \mathrm{dist}_{\measuredangle}(q_u, q_v)$, where $\alpha > 0$ is a weighting parameter and $\mathrm{dist}_{\measuredangle}()$ is the angular distance between two orientations. Sec. V specifies the value of d_{sim}. Condition (i') is shared between RCS and RCS*, while condition (ii") is important for keeping RCS* resolution optimal as it prevents nodes with lower cost from being pruned away by nodes with higher cost.

RCS* uses a set of motion primitives of different resolutions, but instead of applying all motion primitives together, only the coarsest motion primitives are used when a node is initially expanded (line 10). The resolution of a motion primitive (with respect to $\delta\ell$ and $\delta\theta$) is refined when a node with a coarser motion primitive is processed (line 14). More specifically, a finer motion primitive is obtained by changing $\delta\ell$ or $\delta\theta$ by a small value that corresponds to a finer resolution (see [30] for mathematical expressions). Note that resolution refinement is done even if a node is invalid since finer resolutions might be valid.

Resolution refinement will be cut off when reaching a finer resolution, with respect to $\delta\ell$ or $\delta\theta$, than the predefined cutoff resolution $R_{\min}$ (line 15). Here the cutoff resolution is determined by the minimal motions of the needle tip that are precisely achievable with the hardware system.

The algorithm terminates when the OPEN list is exhausted(line 4), and the best plan is returned (if any is found). RCS* is guaranteed to terminate in finite time due to the cutoff resolution.

B. Domain-specific optimization

We describe an additional procedure to further improve RCS*'s performance. We incorporate the concept of inevitable collisions [42] to eliminate potential nodes that would lead to collisions as they are expanded. In particular, for a given vertex v and the goal point, region growing is performed from $\mathbf{x}_v$ within an approximated reachable workspace, considering the existence of obstacles. This region is defined as the intersection of the kinematically forward-reachable workspace and the olive-shaped feasible workspace defined by $\mathbf{x}_v$, p_{goal}, and tolerance τ (see Fig. 2). We mention that due to (i) maximum curvature constraint, (ii) maximum turning angle constraint (the needle would shear or buckle when turning over $\pi/2$), and (iii) maximum insertion length constraint, the kinematically forward-reachable workspace for a given needle configuration is a trumpet-shaped volume (see Fig. 2 left). In the case that the goal is not reached by the grown region, v is discarded. For additional optimizations applicable to RCS*, see [43].

V. Resolution Optimality of RCS*

We study the theoretical properties of RCS* and provide a proof sketch for the algorithm being *resolution optimal*. Informally, resolution optimality implies that RCS* is guaranteed to find a plan whose cost is as close as desired to the cost of the globally optimal qualified motion plan σ^*, assuming that the cutoff resolution $R_{\min} = \{\delta\ell_{\min}, \delta\theta_{\min}\}$ is fine enough. Thm. 1 given below, states our main theoretical contribution relating to the resolution optimality of RCS*.

Before reaching Thm. 1, we introduce the notions of *well-behaved cost*, as well as *robust* and *decomposable* trajectories, which will be used as assumptions on $\mathcal{C}$ and σ^* to prove our result. It may not be possible to approximate $\mathcal{C}$ or σ^* using motion plans with a finite number of transitions without additional constraints on the cost function $\mathcal{C}$ or if the plan σ^* has no clearance from obstacles.

The following definition states that close-by configurations have similar costs and that there are bounds on the values that the cost can attain.

Definition 1 (Well-behaved cost)**.** A configuration-based cost function c is well-behaved if (i) it is Lipschitz continuous, i.e., $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}_{\mathrm{free}}, |c(\mathbf{x}_1) - c(\mathbf{x}_2)| \leq L_c \cdot \rho(\mathbf{x}_1, \mathbf{x}_2)$ for $L_c \in \mathbb{R}$, and (ii) $\forall \mathbf{x} \in \mathcal{X}_{\mathrm{free}}, c(\mathbf{x}) \in [c_{\min}, c_{\max}]$, where $c_{\min}, c_{\max} \in \mathbb{R}$ and $c_{\min} > 0$. In such a case, we also say that the trajectory-based cost function $\mathcal{C}(\sigma) = \int_0^{\ell} c\big(\sigma(s)\big)\mathrm{d}s$ is well-behaved.

We assume $c_{\max}$ is not infinitely large since such configurations can be removed from $\mathcal{X}_{\mathrm{free}}$. We also require a well-behaved cost to satisfy $c_{\min} > 0$ since in the case of needle steering, there is always a cost associated with puncturing tissue, so it makes no sense to allow cost-free regions.

Next, we provide two definitions that are used to characterize motion plans that RCS* can approximate. The first definition (borrowed from [30]) is concerned with trajectories that are induced by a finite set of motion primitives (not necessarily the ones used by RCS*). The second definition is concerned with trajectories that are robust with respect to clearance. A motion plan is considered *qualified* if it satisfies both definitions.

Definition 2 (Decomposable trajectory)**.** A trajectory $\sigma : [0, l] \to \mathcal{X}$ is decomposable if it can be decomposed into a finite set of motion primitives. Namely, there exist primitives $M_\sigma = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\} \subset A$ such that $\sigma = \sigma(0) \oplus M_\sigma$,

where $\mathbf{x} \oplus M$ denotes the resultant trajectory obtained by sequentially applying elements in M to $\mathbf{x}$.

Definition 3 (Robust trajectory)**.** A trajectory $\sigma : [0, l] \to \mathcal{X}$ is δ-robust, for some $\delta > 0$, if it has δ clearance from (i) obstacles $\min_{s\in[0,l],\mathbf{x}\in\mathcal{X}_{\text{obs}}} \rho(\sigma(s), \mathbf{x}) \geq \delta$, and (ii) the goal region boundary, $\|\text{Proj}(\sigma(l)) - p_{\text{goal}}\|_2 \leq \tau - \delta$.

We are ready to state our main theoretical result concerning the resolution optimality of RCS*.

Theorem 1 (Resolution optimality)**.** *Let* $\Delta = (\mathcal{X}, \mathcal{W}_{\text{obs}}, \mathbf{x}_{\text{start}}, p_{\text{goal}}, \tau, \ell_{\max}, \kappa_{\max}, \mathcal{C})$ *be an optimal steerable needle motion-planning problem,* $\varepsilon \in (0, \infty)$ *be an approximation factor, and* σ^* *be a trajectory. Also, suppose that the following conditions are satisfied:*

- ***(C1)*** *The steerable-needle system is Lipschitz continuous;*
- ***(C2)*** *The cost function* $\mathcal{C}$ *is well-behaved and characterized with* $L_c, c_{\min}, c_{\max}$*. Denote* $k = \frac{L_c + c_{\max}}{c_{\min}}$*.*
- ***(C3)*** σ^* *is decomposable and* δ*-robust with* $\delta = \min\{\frac{\varepsilon}{k}, \frac{\tau}{2}\}$*.*
- ***(C4)*** *The radius* d_{sim} *used to reject similar nodes satisfies* $d_{\text{sim}} < \min\left\{ \frac{2}{\kappa_{\max}} \sin \frac{\kappa_{\max}\delta\ell_{\min}}{2}, \frac{\delta(L_s - 1)}{2(L_s^H - 1)} \right\}$, where $H = \left\lceil \frac{\ell_{\max}}{\delta\ell_{\min}} \right\rceil$.

Then RCS is resolution optimal, i.e., for fine-enough cutoff resolution* $R_{\min} = \{\delta\ell_{\min}, \delta\theta_{\min}\}$*, RCS* will find a motion plan that satisfies* $\mathcal{C}(\sigma) \leq (1 + \varepsilon) \cdot \mathcal{C}(\sigma^*)$*.*

A. Proof sketch for Thm. 1

We provide a sketch of the proof for Thm. 1, which consists of two main steps. We first show that the plan σ^* can be approximated by another plan σ^*_β that is composed solely of the motion primitives used by RCS*. Then, we show that even though RCS* might not be able to exactly find σ^*_β due to pruning, it will be able to recover another plan $\tilde{\sigma}^*_\beta$ whose cost is similar to that of σ^*_β (and σ^*). The full proof is in the extended version of our paper [43].

We assume that the conditions in Thm. 1 are met for a plan σ^* and approximation factor $\varepsilon > 0$. As a first step, we show that there exists a resolution $R_{\min}$ and a trajectory σ^*_β that approximates σ^* and is constructed solely from the motion primitives of RCS*. In particular, σ^*_β is a *piecewise β-approximation* of σ^* which is defined as follows: the two trajectories σ^* and σ^*_β can be partitioned into a sequence of trajectories $\sigma_1, \ldots, \sigma_n$ and $\sigma'_1, \ldots, \sigma'_n$, respectively, for some positive and finite integer n, such that for any $1 \leq i \leq n$ (i) the Hausdorff distance between σ_i and σ'_i is at most some value $\beta > 0$, and (ii) the cost of the two trajectories satisfies $\mathcal{C}(\sigma'_i) \leq (1 + \beta)\mathcal{C}(\sigma_i)$ (see formal definition in [43]). This step follows from refining our previous proof that showed the existence of such an approximation, albeit only for the length cost [30, Lemma 2], and relies on the assumptions **C1** and **C2**, which characterize the system and cost function, and assumption **C3**, which states that σ^* is decomposable.

The value β is chosen to guarantee that $\mathcal{C}(\sigma^*_\beta) \leq (1 + \varepsilon)\mathcal{C}(\sigma^*)$. Moreover, the choice of β, and the fact that σ^*_β is a β-approximation of σ^* also ensure that σ^*_β is collision free and satisfies the goal tolerance according to assumption **C3**. Note that the above properties would still hold even if we replace the constant β with a slightly larger value $\beta' > \beta$. However, we use the more conservative value $\beta > 0$ to compensate for the fact that RCS* prunes the tree. In particular, due to pruning, RCS* might eliminate some of the vertices induced by the trajectory σ^*_β, and thus, we cannot guarantee that σ^*_β would be returned as a solution. However, we can show that even in the presence of pruning RCS* will compute a valid plan $\tilde{\sigma}^*_\beta$ that tightly bounds the cost of σ^*_β (and thus tightly bounds the cost of σ^*).

We now elaborate on this. Denote the sequence of motion primitives that define σ^*_β as $M_{\sigma^*_\beta} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$. When the motions in $M_{\sigma^*_\beta}$ are sequentially applied to $\mathbf{x}_{\text{start}}$, we obtain a sequence of configurations $\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_0 = \mathbf{x}_{\text{start}}, \mathbf{x}_i = \mathbf{x}_{i-1} \oplus \mathcal{M}_i, i \in [1, n]$, some of which may be pruned. By carefully bounding d_{sim} (according to **C4**), we guarantee that in the worst case the trajectory obtained by applying the same sequence of motion primitives to pruned nodes stays within a collision-free tunnel around σ^*_β, ensuring that it remains valid. The choice of d_{sim} also takes care of goal tolerance, guaranteeing that the trajectory still satisfies τ goal tolerance. Finally, because (i) pruning is allowed only when there exists a node with equal or smaller cost, and (ii) the subtrajectories obtained from applying the same motion primitive to close-by configurations are strict approximations, the plan $\tilde{\sigma}^*_\beta$ tightly bounds the cost of σ^*_β.

VI. Results

We focus on the medical procedure of lung biopsy for evaluation. Lung and bronchus cancer has the highest death rate among all types of cancer, killing over 130,000 Americans each year [44]. Lung biopsy enables definitive diagnosis of suspicious lung nodules at an early stage, which is important to increase the survival rate. One potential approach to safely and accurately access lung nodules for biopsy and localized treatment is deploying a steerable needle trans-orally through a bronchoscope to avoid transthoracic access which could cause severe side effects [45–48]. In this approach, a physician deploys a bronchoscope through a patient's bronchial tubes and then the steerable needle is deployed through the bronchoscope, exits out of the bronchial tube, and steers in the lung parenchyma to reach the nodule. Our motion planner focuses on the final stage of automatically steering the needle through the lung parenchyma to the nodule, while avoiding anatomical obstacles including large blood vessels, bronchial tubes, and the lung boundary.

We used the method developed in [12] to reconstruct the anatomical workspace from a chest CT scan with the above-mentioned obstacles. We created 100 test cases. For each test case, we randomly sampled a start pose close to the bronchial tubes and a target in the lung parenchyma. We finished creating test cases when the number of test cases reached 100 after rejecting (i) impossible scenarios where the start pose has inevitable collision and (ii) trivial scenarios where the start pose can be connected directly to the goal point with a collision-free arc. The simulated needle has $\kappa_{\max} = (50\text{mm})^{-1}, \ell_{\max} = 100\text{mm}$, and a diameter of 2mm. We set the goal tolerance $\tau = 1.0\text{mm}$.

We compared RCS* in simulation with several planners:

(i) **RRT:** The RRT-based needle planner [10, 15] with 5% goal biasing and 100% goal connecting ratio.

(ii) **AO-RRT:** AO-RRT [22, 23] adapted for steerable needles, with maximum rotation control 2π and a maximum insertion control 20mm. We follow the guidelines in

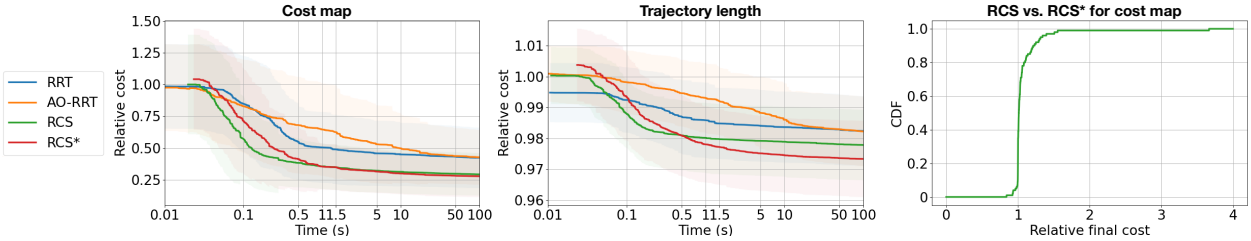


Fig. 3: **Left and middle:** Performance comparison of different methods with two different cost functions. Time axes are logarithmic. The shaded regions show standard deviations. **Right:** A detailed comparison of RCS and RCS* for the cost-map setting, shown as a cumulative distribution function (CDF).

[23] for cost sampling and distance weighting between the configuration space and cost space. For a fair comparison, we use the same goal connecting as RRT.

(iii) **AFT:** The AFT-based needle planner [18, 19], with setup following [19]. AFT internally uses a hybrid cost function; we use $\mathcal{C}_{\text{hybrid}}(\sigma) = \omega \cdot \mathcal{C}(\sigma) + \|\sigma(\ell_\sigma) - p_{\text{goal}}\|_2/\tau$, where ω is a weighting parameter depending on the scale of $\mathcal{C}$. Note that $\mathcal{C}_{\text{hybrid}}$ is only used internally in AFT while $\mathcal{C}$ is always used for performance comparison across different planners.

(iv) **RCS:** The RCS needle planner [30] with cutoff resolution $R_{\min} = \{0.125\text{mm}, 0.157\text{rad}\}$, $\delta\ell_{\max} = 20\text{mm}$.

RCS* used the same basic setup as RCS and $n_{\text{la}} = 3$. Additionally, for all cost functions, we set $\varepsilon = 0.1$. Since we used a CPU implementation of AFT, we did not compare its performance over time, and only reported the final cost after three iterations, as suggested in [18]. For each method, except for AFT, the timeout was set to 100 seconds. All methods except for AFT achieved 100% success rate when timed out while AFT achieved 87% success rate. When reporting the relative cost of AFT, we only consider the test cases that are successfully solved by AFT. As RRT and RCS are designed to find a plan instead of optimizing a plan, we modified them to keep running after the first plan is found, and always return the best plan found when timed out. All experiments were run on a dual 2.1GHz 16-core Intel Xeon Silver 4216 CPU and 100GB of RAM. All parallelizations were implemented with Motion Planning Templates (MPT) [49].

We used two well-behaved cost functions for which RCS* is resolution optimal: (i) trajectory length, and (ii) a cost function informed by a cost map derived from medical images [12], where each voxel in the 3D cost map is associated with a cost value that represents tissue damage. We used trilinear interpolation to smooth out the voxelized cost map and forced $c_{\min} = 0.01$.

Fig. 3 shows results for the different planners for the cost map and trajectory length cost functions. For both cost functions, the cost of a plan may vary significantly between test cases. For example, trajectory length is affected by how far away the target lies relative to the start pose and cost map values are much higher when the needle is steering in a vessel-cluttered region. To account for the large variation between test cases, we first computed *relative cost* within one test case. Specifically, we took the first-found plan (no matter which method returns it as long as it is returned the fastest), and computed relative cost to this plan for all other plans found for the same test case. We did this for all the test cases, and then averaged over the 100 test cases (see the first two plots in Fig. 3). Such relative cost decreased as the result plans are gradually optimized. AFT, which is omitted from those plots, achieved a final cost of 0.574 and 0.991 for cost map and trajectory length, respectively.

For both cost functions, RCS* achieved the best final costs. For the cost map, which is more clinically relevant, RCS* outperformed other methods after 1.5 seconds with a final cost of 0.277, which is 50% lower than AFT (0.574), 35% lower than RRT (0.423) and AO-RRT (0.427), and 7% lower than RCS (0.299). This indicates that the final plan produced by RCS* successfully avoids more small blood vessels than the other methods. It is worth mentioning that unlike trajectory length, small perturbations may lead to very different costs when we use the cost map. As for trajectory length, all methods generated trajectories with roughly similar lengths, although RCS* computed shorter trajectories than all other methods.

In Fig. 3, we report the cumulative distribution function across the 100 test cases of the relative final cost of RCS with respect to RCS* for the cost-map setting. Although RCS achieved comparable costs with RCS* in most cases, for about 20% of the test cases RCS achieved a motion plan that is at least 10% more costly than RCS*. In the extreme case, the final cost of RCS was 3.67 times that of RCS*.

Finally, we mention that the number of nodes RCS* explored is only 53% (for cost map) and 30% (for trajectory length) of that RCS explored. This indicates that although RCS* spent more time choosing the right node to explore, it explores much fewer nodes than RCS to get a plan with even higher quality. The experimental results demonstrate that RCS* is faster than competing methods and that the theoretical guarantees of RCS* have a practical impact on the quality of result plans, which can reduce patient trauma.

VII. Conclusion

In this paper, we introduce the first resolution-optimal motion planner for steerable needles. In particular, our method returns in finite time a motion plan whose cost can be as close as desired to the globally optimal qualified motion plan, assuming the given resolution is fine enough. We also provide a proof sketch to show the resolution optimality of our method with a careful discussion of assumptions and required conditions. We evaluate our proposed planner with simulation experiments and show it can efficiently compute high-quality motion plans considering clinically relevant cost functions. In the future, we plan to investigate speedup techniques for vertex validation and pruning to reduce vertex expansion time. In addition, we plan to develop explicit expressions for the resolution $R_{\min}$ necessary to achieve the desired level of approximation quality for a given problem instance. We will also experimentally evaluate the planner with steerable needles in ex-vivo tissues.

REFERENCES

[1] M. Fu, K. Solovey, O. Salzman, and R. Alterovitz, "steerable-needle-planner," https://github.com/UNC-Robotics/steerable-needle-planner, 2021, accessed: 2022-02-15.

[2] N. Abolhassani, R. Patel, and M. Moallem, "Needle insertion into soft tissue: A survey," *Medical Engineering & Physics*, vol. 29, no. 4, pp. 413–431, 2007.

[3] R. Alterovitz, K. Goldberg, and A. Okamura, "Planning for steerable bevel-tip needle insertion through 2D soft tissue with obstacles," in *IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2005, pp. 1640–1645.

[4] N. J. Cowan, K. Goldberg, G. S. Chirikjian, G. Fichtinger, R. Alterovitz, K. B. Reed, V. Kallem, W. Park, S. Misra, and A. M. Okamura, "Robotic needle steering: design, modeling, planning, and image guidance," in *Surgical Robotics: System Applications and Visions*, J. Rosen, B. Hannaford, and R. M. Satava, Eds. Springer, 2011, ch. 23, pp. 557–582.

[5] W. Park, J. S. Kim, Y. Zhou, N. J. Cowan, A. M. Okamura, and G. S. Chirikjian, "Diffusion-based motion planning for a nonholonomic flexible needle model," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Apr. 2005, pp. 4611–4616.

[6] R. J. Webster III, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *Int. J. Robotics Research (IJRR)*, vol. 25, no. 5-6, pp. 509–525, 2006.

[7] A. Favaro, L. Cerri, S. Galvan, F. Rodriguez y Baena, and E. De Momi, "Automatic optimized 3D path planner for steerable catheters with heuristic search and uncertainty tolerance," in *IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2018, pp. 9–16.

[8] R. Wein, J. Van Den Berg, and D. Halperin, "Planning high-quality paths and corridors amidst obstacles," *Int. J. Robotics Research (IJRR)*, vol. 27, no. 11-12, pp. 1213–1231, 2008.

[9] P. K. Agarwal, K. Fox, and O. Salzman, "An efficient algorithm for computing high-quality paths amid polygonal obstacles," *ACM Transactions on Algorithms (TALG)*, vol. 14, no. 4, pp. 1–21, 2018.

[10] A. Kuntz, L. G. Torres, R. H. Feins, R. J. Webster III, and R. Alterovitz, "Motion planning for a three-stage multilumen transoral lung access system," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3255–3261.

[11] M. P. Strub and J. D. Gammell, "Admissible heuristics for obstacle clearance optimization objectives," *arXiv preprint arXiv:2104.02298v2 [cs.RO]*, 2021.

[12] M. Fu, A. Kuntz, R. J. Webster III, and R. Alterovitz, "Safe motion planning for steerable needles using cost maps automatically extracted from pulmonary images," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4942–4949.

[13] V. Duindam, J. Xu, R. Alterovitz, S. Sastry, and K. Goldberg, "Three-dimensional motion planning algorithms for steerable needles using inverse kinematics," *Int. J. Robotics Research (IJRR)*, vol. 29, no. 7, pp. 789–800, 2010.

[14] K. Hauser, R. Alterovitz, N. Chentanez, A. Okamura, and K. Goldberg, "Feedback control for steering needles through 3D deformable tissue using helical paths," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[15] S. Patil, J. Burgner, R. J. Webster III, and R. Alterovitz, "Needle steering in 3D via rapid replanning," *IEEE Trans. Robotics*, vol. 30, no. 4, pp. 853–864, 2014.

[16] K. M. Seiler, S. P. Singh, S. Sukkarieh, and H. Durrant-Whyte, "Using Lie group symmetries for fast corrective motion planning," *Int. J. Robotics Research (IJRR)*, vol. 31, no. 2, pp. 151–166, 2012.

[17] J. Van Den Berg, S. Patil, R. Alterovitz, P. Abbeel, and K. Goldberg, "LQG-based planning, sensing, and control of steerable needles," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*. Springer, 2010, pp. 373–389.

[18] F. Liu, A. Garriga-Casanovas, R. Secoli, and F. Rodriguez y Baena, "Fast and adaptive fractal tree-based path planning for programmable bevel tip steerable needles," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 601–608, 2016.

[19] M. Pinzi, S. Galvan, and F. Rodriguez y Baena, "The adaptive hermite fractal tree (AHFT): a novel surgical 3D path planning approach with curvature and heading constraints," *Int. J. Computer Assisted Radiology and Surgery*, vol. 14, no. 4, pp. 659–670, 2019.

[20] S. M. LaValle and J. J. K. Jr., "Randomized kinodynamic planning," *Int. J. Robotics Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[21] M. Kleinbort, K. Solovey, Z. Littlefield, K. E. Bekris, and D. Halperin, "Probabilistic completeness of RRT for geometric and kinodynamic planning with forward propagation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. x–xvi, 2018.

[22] K. Hauser and Y. Zhou, "Asymptotically optimal planning by feasible kinodynamic planning in a state–cost space," *IEEE Trans. Robotics*, vol. 32, no. 6, pp. 1431–1443, 2016.

[23] M. Kleinbort, E. Granados, K. Solovey, R. Bonalli, K. E. Bekris, and D. Halperin, "Refined analysis of asymptotically-optimal kinodynamic planning in the state-cost space," in *IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6344–6350.

[24] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robotics Research (IJRR)*, vol. 30, no. 7, pp. 846–894, 2011.

[25] K. Solovey, L. Janson, E. Schmerling, E. Frazzoli, and M. Pavone, "Revisiting the asymptotic optimality of RRT*," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2189–2195.

[26] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *Int. J. Robotics Research (IJRR)*, vol. 35, no. 5, pp. 528–564, 2016.

[27] W. Sun, S. Patil, and R. Alterovitz, "High-frequency replanning under uncertainty using parallel sampling-based motion planning," *IEEE Trans. Robotics*, vol. 31, no. 1, pp. 104–116, 2015.

[28] M. Tsao, K. Solovey, and M. Pavone, "Sample complexity of probabilistic roadmaps via ϵ-nets," in *International Conference on Robotics and Automation*. IEEE, 2020, pp. 2196–2202.

[29] D. Dayan, K. Solovey, M. Pavone, and D. Halperin, "Near-optimal multi-robot motion planning with finite sampling," in *IEEE International Conference on Robotics and Automation*, 2021.

[30] M. Fu, O. Salzman, and R. Alterovitz, "Toward Certifiable Motion Planning for Medical Steerable Needles," in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.

[31] J. Xu, V. Duindam, R. Alterovitz, and K. Goldberg, "Motion planning for steerable needles in 3D environments with obstacles using rapidly-exploring random trees and backchaining," in *IEEE Int. Conf. Automation Science and Engineering*. IEEE, 2008, pp. 41–46.

[32] J. Barraquand and J.-C. Latombe, "Robot motion planning: A distributed representation approach," *Int. J. Robotics Research (IJRR)*, vol. 10, no. 6, pp. 628–649, 1991.

[33] ——, "Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles," *Algorithmica*, vol. 10, no. 2, pp. 121–155, 1993.

[34] P. Cheng and S. M. LaValle, "Resolution complete rapidly-exploring random trees," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, vol. 1. IEEE, 2002, pp. 267–272.

[35] S. R. Lindemann and S. M. LaValle, "Multiresolution approach for motion planning under differential constraints," in *IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2006, pp. 139–144.

[36] D. S. Yershov and S. M. LaValle, "Sufficient conditions for the existence of resolution complete planning algorithms," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*. Springer, 2010, pp. 303–320.

[37] J. D. Gammell and M. P. Strub, "Asymptotically optimal sampling-based motion planning methods," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 295–318, 2021.

[38] R. Shome and L. E. Kavraki, "Asymptotically optimal kinodynamic planning using bundles of edges," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, vol. 30, 2021.

[39] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009.

[40] O. Ljungqvist, N. Evestedt, M. Cirillo, D. Axehill, and O. Holmer, "Lattice-based motion planning for a general 2-trailer system," in *IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 819–824.

[41] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[42] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[43] M. Fu, K. Solovey, O. Salzman, and R. Alterovitz, "Resolution-optimal motion planning for steerable needles," https://robotics.cs.unc.edu/publications/fu2022arXiv.

[44] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, "Cancer statistics, 2022," *CA: a cancer journal for clinicians*, 2022.

[45] A. Kuntz, P. J. Swaney, A. Mahoney, R. H. Feins, Y. Z. Lee, R. J. Webster III, and R. Alterovitz, "Toward transoral peripheral lung access: Steering bronchoscope-deployed needles through porcine lung tissue," in *Hamlyn Symposium on Medical Robotics*, 2016, pp. 9–10.

[46] P. J. Swaney, A. W. Mahoney, B. I. Hartley, A. A. Remirez, E. Lamers, R. H. Feins, R. Alterovitz, and R. J. Webster III, "Toward transoral peripheral lung access: Combining continuum robots and steerable

needles," *Journal of Medical Robotics Research*, vol. 2, no. 01, p. 1750001, 2017.

[47] J. Hoelscher, M. Fu, I. Fried, M. Emerson, T. E. Ertop, M. Rox, A. Kuntz, J. A. Akulian, R. J. Webster III, and R. Alterovitz, "Backward planning for a multi-stage steerable needle lung robot," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3987–3994, 2021.

[48] I. Fried, J. Hoelscher, M. Fu, M. Emerson, T. E. Ertop, M. Rox, J. Granna, A. Kuntz, J. A. Akulian, R. J. Webster III, and R. Alterovitz, "Design considerations for a steerable needle robot to maximize reachable lung volume," in *IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2021.

[49] J. Ichnowski and R. Alterovitz, "Motion planning templates: A motion planning framework for robots with low-power CPUs," in *IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2019, pp. 612–618.