Resolution Optimal Motion Planning for Medical Needle Steering from Airway Walls in the Lung

Janine Hoelscher[1], Inbar Fried[2], Oren Salzman[3], Ron Alterovitz[2]

***Abstract*— Steerable needles are novel medical devices capable of following curved paths through tissue, enabling them to avoid anatomical obstacles and steer to hard-to-reach sites in tissue, including targets in the lung for lung cancer diagnosis. Steerable needles are typically deployed into tissue from an insertion surface, and selecting the insertion site is critical for procedure success as it determines which paths the needle can take to its target. Prior motion planners for steerable needles typically only plan from a specific start pose to the target. We introduce a new resolution-optimal steerable needle motion planner that efficiently finds plans from an insertion surface to a target position, handling additional degrees of freedom at both the start and the target. Our algorithm systematically builds a search tree consisting of needle motion primitives backward from the target towards the insertion surface, which allows it to provide an optimality guarantee up to the resolution of the primitives. The algorithm finds higher-quality plans faster than prior state-of-the-art motion planners, as demonstrated in anatomical scenario simulations in the lung.**

I. Introduction

Steerable needles are minimally invasive devices whose ability to curve allows them to avoid anatomical obstacles and to reach sites in the body not reachable with conventional, straight needles [1]. One application for steerable needles is biopsies for cancer diagnosis. Lung cancer causes more deaths in the United States than any other type of cancer [2]. Early diagnosis requires biopsy and is crucial for favorable patient outcomes. Steerable needles can reach peripheral lung nodules in a minimally invasive way, mitigating patient risk in biopsy procedures.

Safely navigating a steerable needle to a target requires not only computing a feasible obstacle-avoiding path to the target but also determining where the needle should start. The steerable needle lung robot developed by our group [3] delivers a steerable needle into the airways via a bronchoscope. The steerable needle is then inserted through the airway wall before it steers through lung tissue to a biopsy location. Selecting this insertion site on the airway wall is critical for procedure success as it determines which paths the needle can take to its target. Furthermore, for a biopsy procedure, only the target nodule's position is relevant, and no particular target orientation is required. Hence, in contrast to the standard paradigm of planning a collision-free motion from one pose (position and orientation) to another pose, we must compute a motion from an insertion surface to a target position.

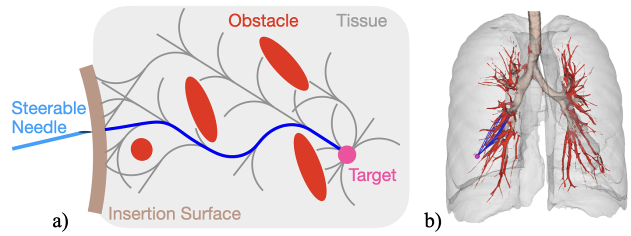

Fig. 1. (a) Steerable needles (light blue) are deployed from an insertion surface (tan) and steer through tissue (grey) towards a target position (pink) while avoiding critical anatomical obstacles. Our new algorithm efficiently finds motion plans (dark blue), providing resolution optimality guarantees by building a search tree backward from the target toward the insertion surface. (b) In a lung biopsy scenario, the steerable needle is deployed from the airways (tan) into lung tissue (grey) while avoiding blood vessels (red) and steering toward a lung nodule (pink).

This work introduces a new motion planner for steerable needles that plans from an insertion surface to a target position, as visualized in Fig. 1. For medical applications, it is crucial to both compute high-quality plans to ensure low patient risk and to find these plans quickly. Our new algorithm is both computationally efficient and resolution-optimal, which means it can find optimal motion plans up to a clinically relevant resolution.

Our new motion planner inverts the planning process and plans backward from a target position to an insertion surface, enabling greater efficiency. In previous work, we introduced a Rapidly Exploring Random Tree (RRT)–based motion planner that employed a similar backward planning strategy [4]. However, RRT-based planners typically do not provide an optimality guarantee, a crucial property for motion planners in critical medical applications. A recent contribution from our group introduced a resolution-optimal motion planner for steerable-needle applications planning from a start pose to a target position [5]. We now expand the resolution-optimal algorithm to compute motion plans from an insertion surface to a target position by incorporating a backward planning strategy that can efficiently handle the additional degrees of freedom.

Our new algorithm systematically builds a search tree backward from the target toward the insertion surface,

*This research was supported by the National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health (NIH) under award R01EB024864, the National Science Foundation (NSF) under award 2008475, and the United States-Israel Binational Science Foundation (BSF) under grant no. 2019703. The content is solely the responsibility of the authors and does not necessarily represent the official views of NIH or NSF.

[1]J. Hoelscher is with the Department of Bioengineering, Clemson University, Clemson, SC 29634, USA. janineh@clemson.edu

[2]I. Fried and R. Alterovitz are with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA.{ifried01, ron}@cs.unc.edu

[3]O. Salzman is with the Department of Computer Science, Technion, Haifa 3200003, Israel osalzman@cs.technion.ac.il

searching for an optimal motion plan. It iteratively expands the tree by adding and refining motion primitives corresponding to the steerable needle's hardware constraints. As no specific orientation for the needle tip at the target is required, we treat the target orientation as a refinement variable and create multiple interleaving search trees. Combined with the backward planning strategy, this allows us to handle the degrees of freedom in both the insertion surface and the target position.

We test our new planning approach in a simulated transoral lung biopsy scenario. Our simulations show that (i) for easy planning scenarios, our new algorithm terminates within three minutes with the resolution-optimal motion plan or a guarantee that no such plan exists, and (ii) for more difficult scenarios, our new algorithm finds lower-cost plans faster than prior sampling-based algorithms. This combination of optimality and efficiency properties is highly beneficial for time-critical medical applications.

II. Related Work

Many steerable needle designs employ base actuation consisting of insertion and axial rotation and an asymmetric (e.g., beveled) needle tip. When such a steerable needle is inserted into tissue, its asymmetric tip applies a force that causes the needle to curve in the direction of the bevel [6]. Steerable needle control strategies can achieve complex 3D trajectories [7], [8], [9]. While this paper is written with bevel-tip steerable needles in mind, the unifying feature of motion planning for steerable needles is their curvature constraint. Thus, our new algorithm could also be adapted to other types of steerable needles [10], [11], [12], [13].

Multiple motion-planning algorithms for steerable needles have been proposed. Sampling-based motion planning algorithms can efficiently find plans in high dimensional spaces [14], [15], [16], [17]. Another type of steerable needle motion planner uses optimization-based strategies [18], [19]. Fractal tree-based approaches systematically search the needle's reachable workspace for motion plans [20], [21]. A deep learning-based strategy employed Generative Adversarial Imitation Learning [22].

Steerable-needle deployment requires insertion into the surface of tissue, but few steerable needle motion planners explicitly take the selection of this start pose into account. Existing strategies include relying on a physician's experience for selecting a start pose [23], exhaustively testing all possible start poses [21], sampling start poses based on the reachability of a needle insertion device [24], or a constrained optimization-based algorithm limited to 2D planning [19]. Planning strategies computing the best start pose for conventional, straight medical instrument insertions exist, but typically rely on geometric computations that are not easily generalizable to account for the curving abilities of steerable needles [25], [26], [27]. In our previous work, we introduced B-RRT, an RRT-based algorithm that plans backward from a target position to an insertion surface but does not provide optimality guarantees [4].

None of the previously developed steerable needle motion planners discussed so far provide completeness or optimality guarantees [28]. One difficulty in motion planning for steerable needles lies in their curvature constraints, making the task of connecting two steerable needle configurations a difficult two-point boundary value problem. While the sampling-based Rapidly Exploring Random Tree (RRT) algorithm [29] is probabilistically complete, the NeedleRRT variant [16] uses modified sampling and node connecting strategies for practical efficiency. Similarly, while the RRT* algorithm is asymptotically optimal, [30], an existing RRT* steerable needle variant requires a smoothing step to ensure compliance with curvature constraints, which changes the optimal path [31]. Resolution optimal motion planners have been introduced for car-like robots with non-holonomic curvature constraints [32], [33], [34]. However, these planners only operate in 2D and the workspaces they cover are much larger relative to their minimum radius of curvature in comparison to a typical steerable needle problem. Our group recently introduced the resolution optimal steerable needle motion planner RCS*, which systematically searches for an optimal plan from a start pose to a target position by building a search tree consisting of iteratively refined motion primitives [5]. In this work, we build on this algorithm to consider an insertion surface while maintaining completeness and optimality guarantees.

III. Problem Definition

We model the steerable needle's kinematics using a 3D unicycle model [6], [35], assuming that the needle follows a constant-curvature arc when inserted. Its reachable workspace is limited by its maximum curvature $\kappa_{\max}$, which is dependent on the needle design and tissue properties such as density. Needle deployment is further limited by its maximum length $\ell_{\max}$ and its axial diameter d_{needle} (to avoid close-by obstacles). Its overall curvature cannot exceed $\pi/2$ as this might cause buckling or sheering through tissue [16]. We denote by $g_{\text{needle}}()$ a function describing the aforementioned constraints, which are satisfied when $g_{\text{needle}}() \geq 0$.

We assume that the steerable needle follows its tip in a follow-the-leader manner [36]. Therefore, we can describe the configuration of the steerable needle by its tip. We denote a steerable needle tip configuration as $\mathbf{x} \in \mathcal{SE}(3) = \mathbb{R}^3 \times \mathcal{SO}(3)$, expressing its position $p(\mathbf{x}) \in \mathbb{R}^3$ and its orientation $q(\mathbf{x}) \in \mathcal{SO}(3)$. We refer to the configuration space of the steerable needle as $\mathcal{X} \subseteq \mathcal{SE}(3)$. Following the kinematics model described above, we model a section of a steerable needle motion plan as a constant-curvature arc $\mathbf{a}_i = \{\mathbf{x}_i, \kappa_i, \ell_i, \theta_i\}$, with curvature κ_i, length ℓ_i, and axial rotation θ_i, connecting configuration $\mathbf{x}_i$ to subsequent configuration $\mathbf{x}_{i+1}$, as visualized in Fig. 2. A steerable needle motion plan is an ordered list of such constant-curvature arc plan sections $\Pi = [\mathbf{a}_1, \dots, \mathbf{a}_i, \dots \mathbf{a}_n]$ of $n \in \mathbb{N}$ plan sections. We define $\mathrm{Shape}(\Pi, \mathrm{s}) : [0, 1] \rightarrow \mathbb{R}^3$ as a mapping of configurations along Π into 3D space.

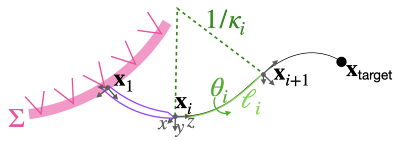

Fig. 2. The steerable needle (green) is deployed from the insertion surface Σ (pink) at configuration $\mathbf{x}_1 \in \Sigma$ and steers towards $\mathbf{x}_{\text{target}}$. A needle plan (black) consists of constant curvature arcs characterized by an insertion length ℓ_i, a curvature κ_i, and a rotation angle θ_i that connect 3D poses $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$.

We assume that the needle operates in a 3D workspace $\mathcal{W} \subset \mathbb{R}^3$ that contains known obstacles $\mathcal{O} \subset \mathcal{W}$ that the needle has to avoid. A configuration $\mathbf{x}$ is said to be collision-free if its position $p(\mathbf{x}) \notin \mathcal{O}$. Similarly, we say that a plan Π is valid if all positions along the plan are collision-free, i.e., $\forall s \in [0, 1], \text{Shape}(\Pi, \text{s}) \notin \mathcal{O}$.

We assume that we are given a cost metric $c : \mathbb{R}^3 \to [0, \infty)$ that assigns a cost to each position $p(\mathbf{x})$ along a plan. Consequently, we define the cost C for a plan Π as

$$C(\Pi) = \int_{s=0}^{1} c(\text{Shape}(\Pi, \text{s}))\text{ds}.$$

Problem 1 [Motion-planning problem] Let $\mathcal{W}$ be a workspace populated with a set of known obstacles $\mathcal{O} \subset \mathcal{W}$, $\Sigma \subset \mathcal{W}$ be an insertion surface, $p_{\text{target}} \in \mathcal{W}$ be a target position, C a cost metric, and $g_{\text{needle}}()$ a function describing the system's kinematic constraints. Our problem calls for finding a motion plan Π^* expressed as the following optimization problem:

$$\Pi^* = \underset{\Pi}{\text{argmin}}\ C(\Pi).$$

Subject to:

$$\begin{aligned} &\forall s \in [0, 1], \text{Shape}(\Pi, \text{s}) \notin \mathcal{O}, \\ &g_{\text{needle}}(\Pi) \geq 0, \\ &p(\mathbf{x}_1) \in \Sigma, \\ &p(\mathbf{x}_n) = p_{\text{target}}. \end{aligned}$$

A unique aspect of our motion-planning problem is the insertion surface $\Sigma \in \mathbb{R}^3$, which corresponds to an insertion surface from which needle deployment begins. Additionally, our problem has an extra degree of freedom at p_{target}, which does not have a fixed orientation.

IV. Methods

This section introduces our new resolution-optimal motion-planning algorithm, BackwardRCS* (B-RCS*). Our approach is an extension of the Resolution Complete Search (RCS*) algorithm [5] that builds and refines a search tree consisting of pre-defined motion primitives. We describe the main features of this algorithm in Sec. IV-A and highlight the changes made in our new approach in Sec. IV-B. Conceptually, the main difference is that RCS* plans from a fixed needle start configuration to a goal point. In contrast, our new algorithm considers an insertion surface consisting of many possible start configurations and determines the optimal one as part of the planning process. To do so, we invert the planning process to begin at p_{target} and plan towards the insertion surface, which requires changing the way the search tree is constructed as well as the target condition. We describe the RCS* algorithm as outlined in Alg. 1. Changes from the RCS* algorithm to the B-RCS* algorithm are highlighted in pink, whereas blue text marks optional speedup strategies explained in Sec. IV-C.

A. Algorithmic Background—RCS*

The main idea behind RCS* [5] is to systematically search the configuration space for an optimal plan by building a search tree consisting of motion primitives that reflect the system's physical constraints. By iteratively refining these motion primitives, an optimality guarantee up to the current refinement level can be given. The algorithm constructs a search tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, where each node $u \in \mathcal{V}$ is associated with a configuration $\mathbf{x}_u$ and each edge $e = (u, v) \in \mathcal{E}$ is associated with a motion plan section $\mathbf{a}_e$ connecting configurations $\mathbf{x}_u$ and $\mathbf{x}_v$. The algorithm associates each search node with a so-called "rank", which is a value that represents both the level of refinement of the motion primitive the node was constructed with and the node's depth in the search tree. The algorithm uses the notion of node rank to determine which nodes to expand next in its search.

The algorithm begins growing the tree from a root node associated with a given start configuration $\mathbf{x}_{\text{start}}$, which in our backward planning implementation is a pose at the target. It maintains a list of nodes called an OPEN list, which contains nodes that have yet to be validated (i.e., that may be in collision or not satisfy the system's constraints). In each iteration, the algorithm selects a node from the OPEN list, tests it for validity, and potentially expands it to create new nodes. This process continues until there are no more nodes left in the OPEN list (Line 5). RCS* selects nodes from the OPEN list based on two criteria, their rank and a secondary metric $f(\cdot)$. This metric is defined as $f(v) = C(v) + h(v)$, where $C(v)$ is the cost associated with the partial plan from the target to node v and $h(v)$ is a heuristic which in our setting is an estimated cost from v to the insertion surface Σ.

The algorithm introduces a lookahead parameter $n_{\text{la}} \in \mathbb{N}$ and sorts all nodes whose rank is $r \leq r_{\text{open}} + n_{\text{la}}$ by their secondary metric, where r_{open} is the lowest rank represented in the OPEN list. The algorithm selects the node with the lowest metric value according to these selection criteria for evaluation. A lower value of n_{la} encourages evaluating coarser resolution nodes first, which can lead to finding an initial plan faster. In contrast, a higher value emphasizes refining the nodes with the lowest metric value.

Additionally, the algorithm employs lazy edge evaluation [37], [38], [39], i.e., it only validates nodes when they are extracted from the OPEN list (Line 7). Node validation includes collision detection along the node's incoming edge. Furthermore, the algorithm verifies that the length ℓ_v of the partial plan from the root to v is shorter than $\ell_{\max}$ and that no nodes similar in position and orientation have already been expanded. If a node passes all validity checks, it is next tested

Algorithm 1 BackwardRCS*

Input: $\mathcal{W}_{\text{obs}}, \Sigma, p_{\text{target}}, \tau, \kappa_{\max}, \ell_{\max}, \delta\ell_{\max}, \delta\phi_{\max}$

1: $\Theta \leftarrow \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}, K \leftarrow \{0, \kappa_{\max}\}$, bestPlan $\leftarrow \emptyset$
2: $\mathbf{x}_{\text{target}} \leftarrow$ InitRoot(p_{target})
3: root $\leftarrow (\mathbf{x}_{\text{target}}, 0)$ ▷ The root has rank 0
4: OPEN $\leftarrow$ {root}, CLOSED $\leftarrow \emptyset$
5: **while** not OPEN.empty() **do**
6: $v \leftarrow$ OPEN.extract()
7: **if** Valid($v, \mathcal{W}_{\text{obs}}, \Sigma, \ell_{\max}$) **then**
8: **if not** existSimilarConfig(v, CLOSED) **then**
9: **if** Reachable($v, \Sigma, \ell_{\max}, \kappa_{\max}$) **then**
10: **if** Terminate(v, Σ, τ) **then**
11: bestPlan.update(v)
12: **if** DirectConnect(v, Σ) **then**
13: bestPlan.update(v)
14: **for** $\mathcal{M} \in$ Primitives($K, \delta\ell_{\max}, \Theta, 0$) **do**
15: OPEN.insert($v \oplus \mathcal{M}$)
16: CLOSED.insert(v)
17: **if** v != root **then**
18: **for** $\mathcal{M} \in$ RefinedPrimitives($\mathcal{M}_v$) **do**
19: OPEN.insert(v.parent $\oplus \mathcal{M}$)
20: **if** v.parent == root **then**
21: OPEN.insert(v.parent $\oplus \mathcal{M}_{\pm\phi}$)
22: **return** bestPlan

for its goal proximity. If the node is closer to p_{target} than a minimum distance $\tau \in \mathbb{R}^+$, the target has been reached, and a complete plan from the start to the target can be retrieved by backtracking the tree structure (Lines 10 and 11).

If the current node has been deemed as valid, but not as reaching the target, the algorithm expands it by the coarsest motion primitives. In this process, the algorithm creates new child nodes and adds them to the OPEN list (Line 15). Additionally, the parent node of the current node v is also expanded. Here, the motion primitives are based on v's current refinement levels. These levels are increased to create refined motion primitives in both path length and rotation, which are then used to expand the parent node (Line 19). This process creates a denser, more refined tree over time. The algorithm terminates when all nodes have been expanded at the finest refinement levels $\delta\ell_{\min}$ and $\delta\theta_{\min}$ and no more nodes are in the OPEN list.

B. The BackwardRCS* (B-RCS*) Algorithm

Our planning problem does not explicitly specify the target orientation, which introduces a new degree of freedom. We suggest discretizing the set of orientations for which we can reach the target and using the RCS* framework to systematically explore *all* possible orientations. To this end, we introduce a new parameter $\delta\phi$, which represents the change in orientation at the target. Similar to RCS*, our new B-RCS* planner operates by refining nodes in the search tree according to their rank (though with a new rank function). The rest of the section formalizes this idea and details additional changes required for our setting.

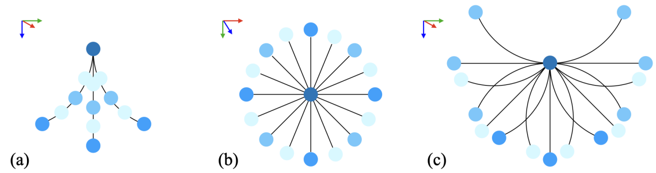

Fig. 3. Motion primitive refinements, lighter colors representing higher-level refinements: (a) insertion length $\delta\ell$, (b) rotation about the insertion direction $\delta\theta$, (c) change in orientation at the root $\delta\phi$.

1) Initial orientation: We set the initial orientation q_{target} such that its deployment direction is $(p_{\text{target}} - p_\sigma)/||p_{\text{target}} - p_\sigma||_2$, pointing toward the position p_σ in the insertion surface Σ that is closest to p_{target} (Line 2). We set the root of the search tree to be the target configuration $\mathbf{x}_{\text{target}} = (p_{\text{target}}, q_{\text{target}})$ (Line 3), starting our backward search from the target and growing the search tree toward Σ.

2) Motion primitives: Recall (Sec. III) that a motion plan is an ordered list of constant-curvature arcs where an arc $\mathbf{a}_i = \{\mathbf{x}_i, \kappa_i, \ell_i, \theta_i\}$ is associated with curvature κ_i, length ℓ_i, and axial rotation θ_i, connecting configuration $\mathbf{x}_i$ to subsequent configuration $\mathbf{x}_{i+1}$. Consequently, we define motion primitives to be a set of parameters that will define the exact set of arcs considered from any configuration $\mathbf{x}$ it reaches.

Formally, we define a motion primitive for our system to be a tuple

$$\mathcal{M} := (\kappa, \delta\ell, \delta\theta, \delta\phi).$$

Here, $\kappa \in [0, \kappa_{\max}]$ is the corresponding arc's curvature (straight or maximum curvature), $\delta\ell$ is the length of the corresponding arc, $\delta\theta$ is the axial rotation (about the insertion direction), and $\delta\phi$ (a new parameter that we add) is the change in orientation at the target.

We define $\delta\phi_{\max} \leq \pi/2$ as the largest deviation in orientation from q_{target} and note that $\delta\phi$ may be non-zero only at the root vertex. An abrupt change in orientation at any other vertex along the plan would defy the steerable needle's curvature constraints. Now, given a motion primitive $\mathcal{M}$, we define $l_\phi(\mathcal{M})$, the orientation's level of refinement to be

$$l_\phi(\mathcal{M}) := \min\{l \in \mathbf{Z}^{\geq 0} | \text{MOD}(\delta\phi, 2^{-l} \cdot \delta\phi_{\max}) = 0\}. \quad (1)$$

Here, MOD($\cdot$) is the modulo operation.

As in RCS* (where the length level l_ℓ and angle levels l_θ are similarly defined), $l_\phi(\mathcal{M})$ is used both in the rank function used to order vertices in OPEN and when refining motion primitives to create new vertices. Specifically, given a search tree vertex v created by applying motion primitive $\mathcal{M}$ from a parent vertex v.parent, we define v's rank to be

$$\text{Rank}(v) := \text{Rank}(v.\text{parent}) + l_\ell(\mathcal{M}_v) + l_\theta(\mathcal{M}_v) + l_\phi(\mathcal{M}_v) + 1.$$

To refine a motion primitive $\mathcal{M} = (\kappa, \delta\ell, \delta\theta, \delta\phi)$ according to the orientation (recall that this is only applied to the root vertex), we construct the following two motion primitives:

$$\mathcal{M}_{\phi\pm} = (\kappa, \delta\ell, \delta\theta, \delta\phi \pm 2^{(l_\phi(\mathcal{M})+1) \cdot \delta\phi_{\max}}).$$

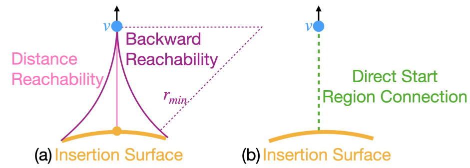

Fig. 4. We speed up the planning process by (a) pruning nodes v for which no part of the insertion surface is within maximum distance ℓ_{needle} (pink) and within its backward reachable workspace (purple) and (b) attempting to directly connect each node chosen for expansion to the insertion surface via a straight connection.

Note that following Eq. 1, both $\mathcal{M}_{\phi+}$ and $\mathcal{M}_{\phi-}$ have orientation level equal to $l_\phi(\mathcal{M})+1$. Finally, in Fig. 3, we visualize the four motion primitive parameters by showing multiple refinements for $\delta\ell$, $\delta\theta$, and $\delta\phi$ (κ is not refined and all possible values are represented in each subfigure).

3) Cost metric & heuristic: Recall that the cost function evaluates an existing partial plan and, therefore, does not have to be changed for planning toward an insertion surface. The heuristic estimate of a plan's cost, however, has to be adapted such that it is a lower bound for any additional cost that occurs between a given node v and the insertion surface Σ. For the path length cost metric, we set $h(v)$ to be the minimum Euclidean distance to any position in the insertion surface. This is a lower bound of the actual cost for any partial plan connecting v and the insertion surface.

4) Termination condition: Let v be a valid search node extracted from OPEN that is not too similar to a previously selected node (Lines 7 and 8). Let $p(\mathbf{x}_v)$ be the position at v and $p_\Sigma(\mathbf{x}_v)$ be the closest point to $p(\mathbf{x}_v)$ on Σ. Given a threshold parameter τ, v is considered to reach Σ if $\tau \geq ||p(\mathbf{x}_v) - p_\Sigma(\mathbf{x}_v)||_2$. This condition is checked in Line 10. If it holds, a plan is retrieved (Line 11).

5) Resolution-Optimality Proof Sketch: The original RCS* algorithm is optimal up to the finest resolution of refined motion primitives expanded [5]. The additional motion primitive $\delta\phi$ in our new B-RCS* algorithm represents the orientation at the target. Conceptually, B-RCS* creates multiple interleaving search trees starting from the target when $\delta\phi$ is refined. In combination with the other motion primitives, these refinements result in covering the complete search space up to the resolution of $\delta\phi$. We also adapted cost heuristic $h(v)$ to estimate a lower bound between node v and any position on the insertion surface, which is required for an optimal search.

C. Speedup Strategies

RCS* employs several speedup strategies, including parallelization, node pruning, direct-goal connecting, and inevitable-collision detecting [5]. We introduce two additional speedup strategies that are particular to our new B-RCS* algorithm.

1) Start reachability pruning: We can test if it is possible to reach the insertion surface Σ from a given vertex v based on the steerable needle's hardware constraints and prune the node if this is not the case (Line 9). This corresponds to testing for distance constraints and curvature constraints.

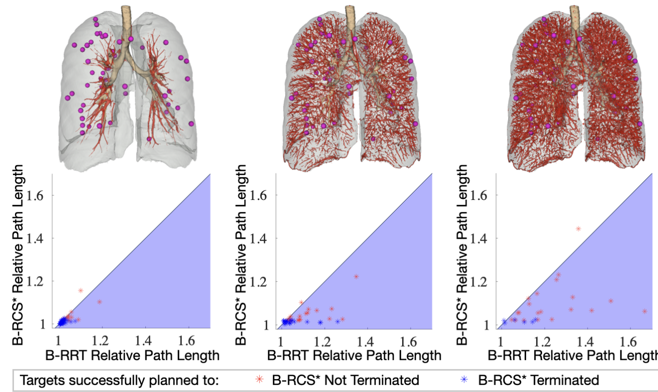

Fig. 5. In three lung planning scenarios with randomly sampled targets (pink) and an increasing obstacle density (red) we compare the performance of B-RCS* to B-RRT by measuring the shortest relative path lengths found for each target. Dots in the lavender-shaded area represent planning scenarios for which B-RCS* found shorter plans than B-RRT. For easier scenarios, B-RCS* terminates with the optimal plan (blue), whereas for harder scenarios, B-RCS* finds plans on average shorter than B-RRT (red).

Specifically, let $d_{\text{min}} = \min_{p_\sigma \in \Sigma} ||p(\mathbf{x}_v) - p_\sigma||_2$ be the minimal distance from v to Σ. If $d_{\text{min}} + \ell_v > \ell_{\text{max}}$, then any path connecting v to Σ will be longer than the steerable needle's total length, and v can be pruned.

Similarly, we test if any position on Σ is in the backward reachable workspace of v. This reachable workspace is limited by the maximum curvature κ_{max} and the total curvature limit of $\pi/2$ to avoid sheering. These two constraints result in a trumpet-shaped reachable workspace, as shown in Fig. 4(a).

2) Direct insertion-surface connection: We further speed up the planning process by attempting to connect the current node v to Σ with a straight-line connection. This strategy is based on the observation that when Σ is large, such straight-line connections often yield feasible plans. This strategy is motivated by goal biasing in RRTs, which has a similar effect in motivating tree growth towards the target [40]. We apply this speedup strategy after having determined that v does not already reach the insertion surface (Line 12).

We determine the current node's configuration $\mathbf{x}_v$ and extend a new straight plan section in its deployment direction up to a total plan length of ℓ_{max} while checking for collisions, as visualized in Fig. 4(b). A new valid motion plan has been found if the extension reaches the insertion surface Σ. However, if such a collision-free connection is determined, it is not necessarily part of the lowest-cost plan. Therefore, we continue the search by expanding v as previously described in case it might contribute to a better plan.

V. Evaluation

We tested our new algorithm in a transoral lung biopsy planning scenario. We used anatomy representations from an existing dataset [41], treating anatomical features, including large blood vessels and the lung boundary, as obstacles and the airway walls as the insertion surface, as shown in Fig. 5. Lungs contain a lot of small blood vessels, which are not always avoidable during procedures. By varying the blood vessel threshold used in image segmentation [42], we varied the size of blood vessels considered significant; a higher

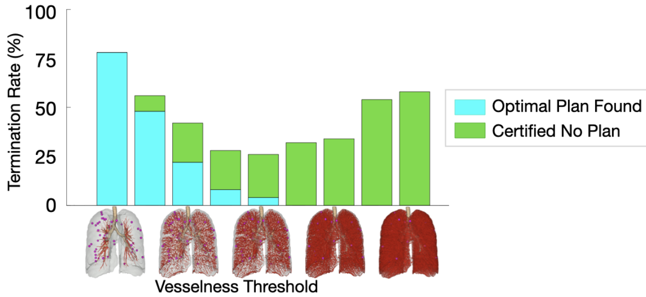

Fig. 6. Termination rates of B-RCS* across scenarios with a varying density of blood vessel obstacles. B-RCS* terminates with a resolution-optimal plan (cyan) more frequently in lower-density scenarios or by certifying that no plan exists (green) more often in higher-density scenarios.

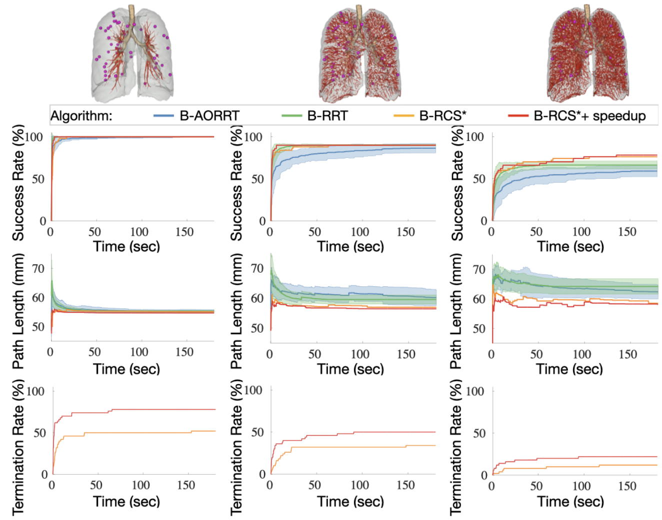

Fig. 7. Simulation results over time recording the success rate to find at least one plan per target, the shortest plan found, and the rate of successful termination with a resolution-optimal plan.

blood vessel obstacle density results in obstacles taking up a greater percentage of the workspace, resulting in a more difficult planning problem. We used a nearest neighbor data structure [43] for voxel-level collision detection and a C++-based planning library [44] for efficient algorithm implementation. We ran all simulations on a 2.9GHz 24-thread Intel Core i9-8950HK CPU with 32GB of RAM.

We compared our new B-RCS* planner with a sampling-based backward needle RRT (B-RRT) planner [4] and an Asymptotic Optimality RRT [45] adapted to steerable needle backward planning (B-AORRT). We compared their performance using the shortest path metric on 50 randomly sampled targets within the free space of the lung, and we ran each planning simulation for 3 min. We set the minimum resolution of B-RCS* to $\delta\ell_{\text{min}} = 0.125$mm, $\delta\theta_{\text{min}} = 0.157$rad, and $\delta\phi_{\text{min}} = 0.157$rad. Based on experimental data [46], we set the steerable needle's minimum radius of curvature to $r_{\text{min}} = 100$mm and the maximum deployment length to $l_{\text{needle}} = 100$mm.

In three lung planning scenarios with increasing density of blood vessel obstacles, we compared the performance of B-RCS* with B-RRT and B-AORRT for each planning target, as shown in Fig. 5. We determined the relative shortest path lengths by dividing the lengths of all plans found by the shortest Euclidean distance between the insertion surface and the respective target position. This relative path length is a proxy for planning difficulty, as longer relative plans indicate more obstacle avoidance necessary in finding a plan. Only targets for which both algorithms found at least one plan are shown. The B-RCS* algorithm terminated with a guaranteed resolution-optimal result for targets with shorter relative path lengths. For targets with longer relative path lengths, while not terminating, it found shorter plans than the B-RRT. This trend can be seen across all three scenario difficulty levels.

To further analyze B-RCS*'s termination behavior, we tested it in eight lung scenarios with increasing density of blood vessel obstacles. We recorded for each scenario how many targets the planner terminated for. Termination can occur because the resolution-optimal solution was found, and it is guaranteed that no better plan exists, as analyzed above in Fig. 5. Further, it can terminate when it asserts that no plan exists. Fig. 6 shows the distribution of terminated planner runs. For easier scenarios, the number of terminations with an optimal plan was higher. With an increasing density of obstacles, the number of runs guaranteeing that no plan can be found up to the minimum resolution increased. Termination is a desirable feature for medical procedure planning as it provides a guarantee that the safest plan for a procedure has been found or that no safe plan exists.

We also compared B-RCS* with B-RRT and B-AORRT as a function of the running time as shown in Fig. 7. For B-RRT and B-AORRT, we repeated the planning process 10 times for each target to account for their random sampling component. B-RCS* found shorter plans faster to more targets than B-RRT and B-AORRT across all lung scenarios, and the speedup strategies further increased B-RCS*'s performance. B-RCS* terminated early for a significant number of targets after having found the resolution-optimal plan. In contrast, B-RRT and B-AORRT run in an anytime fashion and do not terminate. The difference between the algorithms was more pronounced in the more difficult scenarios with a higher density of blood vessel obstacles.

VI. Conclusion

In this work, we introduced a new motion planner that creates plans for steerable needles from an insertion surface to a target position. Its systematic strategy for building a search tree based on motion primitives ensures resolution optimality with respect to the cost metric chosen. Our planning simulations for a lung biopsy task showed that it is more efficient than other sampling-based approaches that provide no such guarantees. Ensuring optimality while creating motion plans in a timely manner is important, especially when planning during a medical procedure.

We plan to develop a rapid replanning routine that leverages the search tree computed during the initial planning process to correct for potential uncertainty during steerable needle deployment. We plan to test this new algorithm in different anatomical scenarios and — most importantly — in physical ex vivo and in vivo experiments.

REFERENCES

[1] 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*. Springer, 2011, pp. 557–582.

[2] American Cancer Society, "Cancer facts and figures," *American Cancer Society Tech. Rep.*, 2022.

[3] A. Kuntz, M. Emerson, T. E. Ertop, I. Fried, M. Fu, J. Hoelscher, M. Rox, J. Akulian, E. A. Gillaspie, Y. Z. Lee *et al.*, "Autonomous medical needle steering in vivo," *Science Robotics*, vol. 8, no. 82, p. eadf7614, 2023.

[4] 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," *Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 3987–3994, 2021.

[5] M. Fu, K. Solovey, O. Salzman, and R. Alterovitz, "Toward certifiable optimal motion planning for medical steerable needles," *The International Journal of Robotics Research*, vol. 42, no. 10, pp. 798–826, 2023.

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

[7] T. E. Ertop, M. Emerson, M. Rox, J. Granna, F. Maldonado, E. A. Gillaspie, M. Lester, A. Kuntz, D. C. Rucker, M. Fu, J. Hoelscher, I. Fried, R. Alterovitz, and R. J. Webster, III, "Steerable needle trajectory following in the lung: Torsional deadband compensation and full pose estimation with 5dof feedback for needles passing through flexible endoscopes," in *Dynamic Systems and Control Conference*, vol. 84270. American Society of Mechanical Engineers, 2020, p. V001T05A003.

[8] D. S. Minhas, J. A. Engh, M. M. Fenske, and C. N. Riviere, "Modeling of needle steering via duty-cycled spinning," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 2756–2759.

[9] D. C. Rucker, J. Das, H. B. Gilbert, P. J. Swaney, M. I. Miga, N. Sarkar, and R. J. Webster, "Sliding mode control of steerable needles," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1289–1299, 2013.

[10] R. Secoli and F. Rodriguez y Baena, "Adaptive path-following control for bio-inspired steerable needles," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2016, pp. 87–93.

[11] T. K. Adebar, J. D. Greer, P. F. Laeseke, G. L. Hwang, and A. M. Okamura, "Methods for improving the curvature of steerable needles in biological tissue," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1167–1177, 2015.

[12] A. J. Sperry, T. J. Schwehr, E. K. Pinegar, O. B. Richards, J. D. Rolston, M. D. Alexander, B. Coats, J. J. Abbott, and A. Kuntz, "Screw-tip soft magnetically steerable needles," *IEEE Transactions on Medical Robotics and Bionics*, vol. 6, no. 1, pp. 4–17, 2024.

[13] M. Babaiasl, F. Yang, S. Boccelli, and J. P. Swensen, "Fracture-directed waterjet needle steering: Design, modeling, and path planning," in *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2020, pp. 1166–1173.

[14] C. Caborni, S. Y. Ko, E. De Momi, G. Ferrigno, and F. Rodriguez y Baena, "Risk-based path planning for a steerable flexible probe for neurosurgical intervention," in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2012, pp. 866–871.

[15] A. Favaro, A. Segato, F. Muretti, and E. De Momi, "An evolutionary-optimized surgical path planner for a programmable bevel-tip needle," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1039–1050, 2021.

[16] S. Patil, J. Burgner, R. J. Webster, and R. Alterovitz, "Needle steering in 3-d via rapid replanning," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 853–864, 2014.

[17] 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 *2008 IEEE international conference on automation science and engineering*. IEEE, 2008, pp. 41–46.

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

[19] R. Alterovitz, K. Goldberg, and A. Okamura, "Planning for steerable bevel-tip needle insertion through 2d soft tissue with obstacles," in *Proc. International Conference on Robotics and Automation (ICRA)*. IEEE, 2005, pp. 1640–1645.

[20] 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.

[21] 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," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 4, pp. 659–670, 2019.

[22] A. Segato, M. Di Marzo, S. Zucchelli, S. Galvan, R. Secoli, and E. De Momi, "Inverse reinforcement learning intra-operative path planning for steerable needle," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 6, pp. 1995–2005, 2021.

[23] A. Segato, F. Calimeri, I. Testa, V. Corbetta, M. Riva, and E. De Momi, "A hybrid inductive learning-based and deductive reasoning-based 3-d path planning method in complex environments," *Autonomous Robots*, vol. 46, no. 5, pp. 645–666, 2022.

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

[25] M. Feng, X. Jin, W. Tong, X. Guo, J. Zhao, and Y. Fu, "Pose optimization and port placement for robot-assisted minimally invasive surgery in cholecystectomy," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 4, p. e1810, 2017.

[26] Y. Hayashi, K. Misawa, and K. Mori, "Optimal port placement planning method for laparoscopic gastrectomy," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 10, pp. 1677–1684, 2017.

[27] L. W. Sun and C. K. Yeung, "Port placement and pose selection of the da Vinci surgical system for collision-free intervention based on performance optimization," in *Proc. International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2007, pp. 1951–1956.

[28] O. Salzman, "Sampling-based robot motion planning," *Commun. ACM*, vol. 62, no. 10, pp. 54–63, 2019.

[29] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *TR 98-11, Computer Science Dept., Iowa State Univ.*, 1998.

[30] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[31] 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 *Proc. International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 9–16.

[32] J. Barraquand and J.-C. Latombe, "Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles," *Algorithmica*, vol. 10, no. 2, pp. 121–155, 1993.

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

[34] 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.

[35] 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. International Conference on Robotics and Automation (ICRA)*. IEEE, 2005, pp. 4600–4605.

[36] H. Choset and W. Henning, "A follow-the-leader approach to serpentine robot motion planning," *Journal of Aerospace Engineering*, vol. 12, no. 2, pp. 65–73, 1999.

[37] B. J. Cohen, M. Phillips, and M. Likhachev, "Planning single-arm manipulations with n-arm robots," in *Robotics: Science and Systems*, D. Fox, L. E. Kavraki, and H. Kurniawati, Eds., 2014.

[38] N. Haghtalab, S. Mackenzie, A. D. Procaccia, O. Salzman, and S. S. Srinivasa, "The provable virtue of laziness in motion planning," in *International Conference on Automated Planning and Scheduling*, M. de Weerdt, S. Koenig, G. Röger, and M. T. J. Spaan, Eds., 2018, pp. 106–113.

[39] A. Mandalika, O. Salzman, and S. S. Srinivasa, "Lazy receding horizon A* for efficient path planning in graphs with expensive-to-evaluate edges," in *International Conference on Automated Planning and Scheduling*. AAAI Press, 2018, pp. 476–484.

[40] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[41] I. Fried, J. Hoelscher, J. A. Akulian, and R. Alterovitz, "A dataset of anatomical environments for medical robots: Modeling respiratory deformation," *arXiv preprint arXiv:2310.04289*, 2023.

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

[43] J. Ichnowski and R. Alterovitz, "Concurrent nearest-neighbor searching for parallel sampling-based motion planning in SO(3), SE(3), and euclidean spaces," *Springer*, 2018.

[44] ——, "Motion planning templates: A motion planning framework for robots with low-power CPUs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 612–618.

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

[46] M. Rox, M. Emerson, T. E. Ertop, I. Fried, M. Fu, J. Hoelscher, A. Kuntz, J. Granna, J. Mitchell, M. Lester, F. Maldonado, E. A. Gillaspie, J. A. Akulian, R. Alterovitz, and R. J. Webster, "Decoupling steerability from diameter: Helical dovetail laser patterning for steerable needles," *IEEE Access*, vol. 8, pp. 181 411–181 419, 2020.