Interactive-rate Motion Planning for Concentric Tube Robots

Luis G. Torres, Cenk Baykal, and Ron Alterovitz

Abstract— Concentric tube robots may enable new, safer minimally invasive surgical procedures by moving along curved paths to reach difficult-to-reach sites in a patient's anatomy. Operating these devices is challenging due to their complex, unintuitive kinematics and the need to avoid sensitive structures in the anatomy. In this paper, we present a motion planning method that computes collision-free motion plans for concentric tube robots at interactive rates. Our method's high speed enables a user to continuously and freely move the robot's tip while the motion planner ensures that the robot's shaft does not collide with any anatomical obstacles. Our approach uses a highly accurate mechanical model of tube interactions, which is important since small movements of the tip position may require large changes in the shape of the device's shaft. Our motion planner achieves its high speed and accuracy by combining offline precomputation of a collision-free roadmap with online position control. We demonstrate our interactive planner in a simulated neurosurgical scenario where a user guides the robot's tip through the environment while the robot automatically avoids collisions with the anatomical obstacles.

I. INTRODUCTION

Concentric tube robots are tentacle-like robotic devices designed for minimally invasive surgery. Their curving ability and small size allow them to reach anatomical sites inaccessible to traditional, straight surgical instruments. Concentric tube robots may enable new, safer surgical access to many sites in the human body, including the skull base [1], the lungs [2], and the heart [3].

These robots are composed of thin, pre-curved, elastic tubes that are nested within one another. The device's maneuverability is enabled via telescopically inserting and rotating each tube, causing the entire robot's shape to change. This powerful shape-changing property also poses a major challenge: unintuitive kinematics determined by mechanical interactions between the device's curved elastic tubes. A physician would therefore find it nearly impossible to safely and accurately guide the robot to perform a surgical task by manually rotating and inserting each tube.

We look to computation to enable intuitive guidance by a physician. Kinematic modeling of concentric tube robots has made great strides recently, allowing for enough speed and accuracy in shape computation to achieve interactive position control of the robot's tip [5], [6], [7]. However, these methods do not account for obstacles. Collisions with anatomical

L. G. Torres, C. Baykal, and R. Alterovitz are with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27517, USA {luis,baykal,ron}@cs.unc.edu
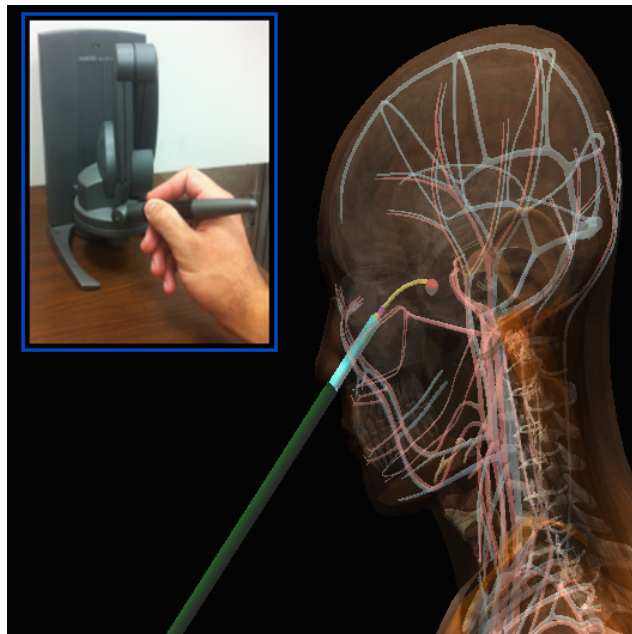
Fig. 1. Virtual simulation of a concentric tube robot being controlled with a SensAble Phantom Desktop [4]. Our interactive motion planner enables the user to move the robot's tip while ensuring that the robot's entire shaft avoids contact with known sensitive anatomical structures.

obstacles can increase risk to the patient and can bend the device unpredictably, impeding effective control. Requiring a physician to enforce collision avoidance when using a position control interface places a significant burden on the physician. Furthermore, even if the physician successfully steers the tip clear of obstacles, reaching for a given tip position can cause a dramatic change in the robot's shape, potentially causing collision of the robot with anatomical obstacles.

We present a motion planner that computes collision-free plans for concentric tube robots at interactive rates. We assume a pre-operative image (e.g., CT scan or MRI) is obtained prior to the procedure, as is common for surgery. From these images anatomical obstacles can be segmented [8]. Our interactive-rate motion planner can potentially allow a physician to continuously specify a desired tip location for the concentric tube robot using a 3D mouse (e.g., a SensAble Phantom [4]), and the robot can interactively respond by reaching the desired tip position while ensuring that the entire device shaft avoids anatomical obstacles.

Achieving interactive-rate motion planning for concentric tube robots is complicated by their kinematics. Accurately evaluating the robot's kinematics is critical for obstacle

avoidance. However, accurately estimating the robot's shape requires solving a numerical system [9], [10] that is sufficiently computationally expensive that previous motion planners would be rendered too slow to be used interactively during a procedure.

In this paper, we achieve interactive rates by creating a motion planner specifically designed for concentric tube robots that mixes precomputation and position control. In our sampling-based motion planning approach, we begin by precomputing a roadmap of collision-free paths in the robot's configuration space, and then search for paths on this roadmap during the procedure. We then use a position control method based on iterative inverse kinematics (IK) to reach user-specified positions not exactly represented in the precomputed roadmap. To link the sampling-based and control approaches, the precomputed roadmap caches shape information that is computationally expensive to compute online, speeding up the iterative IK. This results in a method that quickly computes collision-free motion plans to a region of interest, and then uses fast position control to locally guide the robot tip closer to the position specified by the physician.

We demonstrate our new interactive-rate motion planner in a simulated neurosurgical scenario where a user specifies 3D positions via a 3D mouse and the planner interactively computes control input trajectories to reach the specified points while avoiding collisions with anatomical obstacles.

II. RELATED WORK

In order to accurately compute the shape of the concentric tube robot, we need an accurate kinematic model. Kinematic modeling of concentric tube robots has rapidly improved in recent years, from torsionally rigid models [11], to torsionally compliant models [10], [9], to models that consider external loading [12], [13]. In this paper we use a highly accurate model developed by Rucker et al. [6].

For our application, we are interested in planning collision-free motions fast enough to interactively follow user inputs. Rucker et al. and Xu et al. achieved fast tip position control by quickly computing the manipulator Jacobian and using damped least squares (DLS) IK [6], [7]. These position control methods do not take obstacles into account and may fail to converge to the desired goal position. Dupont et al. produced fast tip position control by approximating the robot's kinematics with a Fourier series and using root finding on this approximation to quickly evaluate IK [5]. This approach feasibly allows for obstacle avoidance, but it does not benefit from the robot's redundant degrees of freedom to consider alternate configurations for reaching goals.

To compute collision-free motion plans for concentric tube robots, early work assumed simplified kinematic models for fast performance [14], [15]. Torres et al. integrated a mechanically accurate model of concentric tube robots into a motion planner by combining a sampling-based roadmap with Jacobian-based goal biasing [16]. However, the computation time required for accurate robot shape computation makes this approach too slow for interactive motion planning.

A related domain where interactive-rate planning is required is for dynamic environments, where motion plans must be computed quickly enough to remain valid under movement of obstacles. Approaches include re-use of paths from prior planning iterations [17], [18], lower-dimensional grid searches to guide the full-dimensional search [19], and repairing paths precomputed in a static environment [20]. In this paper, we assume static obstacles, but we are hampered by kinematics computations that are too expensive for the online planning and collision detection required by most previous replanning methods. Our approach is most similar to that of van den Berg et al. [20] in that we precompute a collision-free roadmap offline to be used for online planning, but we also add concentric tube robot shape caching and iterative IK.

III. PROBLEM FORMULATION

A. Kinematic Modeling

We consider a concentric tube robot with N telescoping tubes numbered in order of increasing cross-sectional radius. Each tube i consists of a straight segment of length l_s^i followed by a pre-curved portion of length l_c^i and constant radius of curvature r_i. The device is inserted at a point $\mathbf{x}_{\text{start}}$ along a vector $\mathbf{v}_{\text{start}}$. We assume that the start location $\mathbf{x}_{\text{start}}$ and initial orientation $\mathbf{v}_{\text{start}}$ correspond to the orifice through which the device is deployed based on the clinical procedure.

Each tube may be (1) inserted or retracted from the previous tube, and (2) axially rotated, yielding 2 degrees of freedom per tube. Hence, the robot's configuration is a $2N$ dimensional vector $\mathbf{q} = (\theta_i, \beta_i : i = 1, \dots, N)$ where θ_i is the axial angle at the base of the i'th tube and $\beta_i < 0$ is the arc-length position of the base of tube i behind the robot insertion point $\mathbf{x}_{\text{start}}$, where $\mathbf{x}_{\text{start}}$ corresponds to the arc-length value 0. The configuration space is $\mathcal{Q} = (S^1)^N \times \mathbb{R}^N$.

For a given configuration $\mathbf{q} \in \mathcal{Q}$, we represent the device's shape as $\mathbf{x}(\mathbf{q}, s) : \mathbb{R}^{2N} \times \mathbb{R} \mapsto \mathbb{R}^3$. The function $\mathbf{x}$ is a 3D space curve parameterized by $s \in [0, 1]$. We note $\mathbf{x}(\mathbf{q}, 0) = \mathbf{x}_{\text{start}}$ and $\mathbf{x}(\mathbf{q}, 1)$ maps to the 3D position of the tip of the robot. To estimate the device's shape $\mathbf{x}$ we use the mechanics-based model developed by Rucker et al. [6].

B. Interactive-rate Motion Planning Problem Formulation

Our interactive-rate motion planner is intended for use by a physician who operates the robot by continuously specifying a new goal position $\mathbf{x}_{\text{goal}}$ for the robot's tip. Our approach repeatedly formulates a new motion planning problem using the latest $\mathbf{x}_{\text{goal}}$. Each problem is to compute a plan Π, denoted by a sequence of configurations $\Pi = (\mathbf{q}_1, \dots, \mathbf{q}_n)$, to move the robot from its current configuration $\mathbf{q}_1$ to a new configuration such that the robot's tip $\mathbf{x}(\mathbf{q}_n, 1)$ is at the given $\mathbf{x}_{\text{goal}}$. The plan must avoid anatomical obstacles, satisfy kinematic constraints, be intuitive to the physician, and should be computed sufficiently fast to enable interactivity.

Anatomical obstacles can be defined in any geometric representation that enables computation of a predicate function `isCollisionFree(q)` that returns `true` if the robot is free of collisions at configuration $\mathbf{q}$ and `false` otherwise. In our

work we consider obstacles defined by 3D polygonal meshes, which can be generated from medical image segmentations [8].

In addition to requiring obstacle avoidance, we must also constrain the set of valid tube insertion values $\boldsymbol{\beta} = (\beta_i | 1 \leq i \leq N)$ due to limitations of the robot's physical design. The carriers that grasp the tube bases have thickness δ; they move on a track of length l_{track}; they cannot move past one another; and our kinematic model [6] requires that the inserted length of tubes increase with decreasing cross-sectional radius. Hence, a configuration is considered feasible only if the following hold (for $N \geq 2$):

$$- l_{\text{track}} \leq \beta_1,$$
$$\beta_{i-1} + \delta \leq \beta_i \quad \text{for } i = 2, \ldots, N,$$
$$\beta_{i-1} + l_s^{i-1} + l_c^{i-1} \geq \beta_i + l_s^i + l_c^i \quad \text{for } i = 2, \ldots, N, \quad (1)$$
$$\beta_N < 0, \text{ and}$$
$$\beta_N + l_s^i + l_c^i \geq 0.$$

Since the motion planner is being executed interactively, the user may specify a goal position $\mathbf{x}_{\text{goal}}$ that is unreachable or for which a plan cannot be computed due to kinematic constraints and the presence of obstacles. Our objective is to compute a plan Π^* that minimizes the tip distance to the target, i.e., minimize $\|\mathbf{x}(\mathbf{q}_n, 1) - \mathbf{x}_{\text{goal}}\|$.

To reach $\mathbf{q}_n$, our method typically considers multiple routes from $\mathbf{q}_1$. For intuitive motion, we select the plan for which the distance traveled by the robot's tip is smallest. In some cases, moving from $\mathbf{q}_1$ to $\mathbf{q}_n$ cannot be done by moving the tip along a straight line from $\mathbf{x}(\mathbf{q}_1, 1)$ to $\mathbf{x}(\mathbf{q}_n, 1)$ without the device's shaft colliding with obstacles; consequently, our method may at times partially retract the device and then redeploy to the new goal. Such non-local motions may be undesirable in some clinical scenarios, e.g., suturing. However, such motions are often necessary and desirable (e.g., reaching a site for biopsy), and are available to the physician using our approach.

IV. Interactive-rate Motion Planning

Our motion planning approach blends sampling-based roadmap planning (for global routing through the robot's free configuration space) with an iterative IK solver (which makes local adjustments for high accuracy).

Our method proceeds in two phases: a *precomputation phase* followed by an *interactive planning phase*, as shown in Fig. 2. In the precomputation phase, we generate a roadmap using a variant of the rapidly exploring random graph (RRG) [21], which we describe in Sec. IV-A. We cache information in the graph data structure to facilitate interactive performance during the actual procedure. In the interactive planning phase, we use the roadmap combined with iterative IK to compute a plan from the robot's current configuration to the new desired tip position specified using a 3D mouse.

A. Precomputation Phase

The input to the precomputation phase is a representation of the anatomy (for collision detection), the design parameters of the robot (to accurately compute kinematics), and
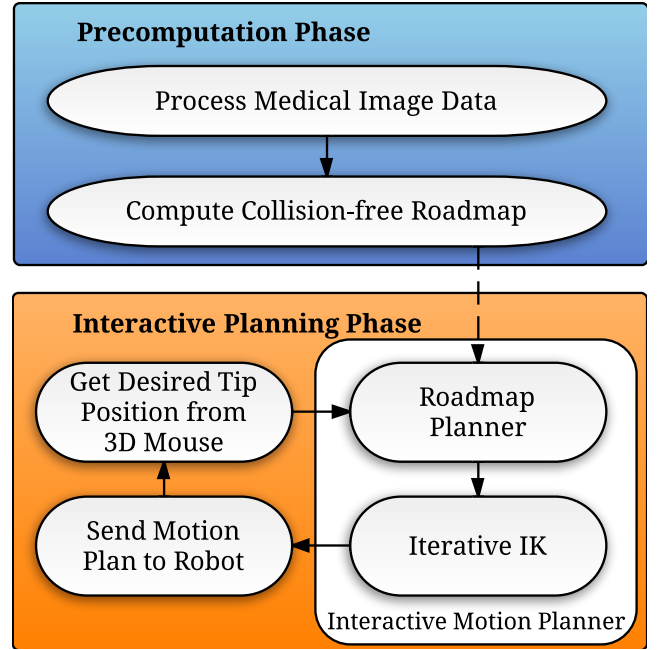

Fig. 2. Method overview.

the robot's insertion location and direction. The output of the precomputation phase is a precomputed roadmap with cached data that will facilitate interactive planning.

In our system, we precompute one roadmap and use it for multiple motion planning queries during the procedure. The roadmap G consists of a set of nodes V corresponding to collision-free robot configurations and a set of directed edges E encoding collision-free motions between the configurations. Since anatomical environments relevant to concentric tube robot robots are highly constrained, large portions of the configuration space are not reachable from a given insertion point. Hence, we grow our roadmap from a given starting state using a modified RRG.

The RRG algorithm builds a roadmap by beginning at a given starting configuration, and only adds configuration samples to the graph that can be connected by a collision-free motion to the roadmap. RRG also refines the roadmap by adding edges in a way that guarantees *asymptotic optimality* of plan quality under certain assumptions. After executing our customized RRG algorithm for some number of iterations, the result is a roadmap that can be quickly queried for high-quality collision-free plans between any two configurations in the roadmap. We then save the roadmap to the hard disk in a format that can be quickly reloaded for use during the surgical procedure. Next, we describe our customizations to RRG that enable fast motion planning for concentric tube robots.

1) Distance Metrics for Roadmap Construction: The canonical RRG algorithm uses a given distance metric $d : \mathcal{Q} \times \mathcal{Q} \mapsto \mathbb{R}$ for (1) sampling a new configuration $\mathbf{q}$ to add to the roadmap, i.e., roadmap *expansion*; and (2) selecting a set of configurations Q_{near} near $\mathbf{q}$ to check for collision-free connections, i.e., roadmap *refinement*. In our

roadmap generation method, we use two distinct distance (pseudo)metrics for each of these roadmap operations.

For roadmap expansion, we use a weighted Euclidean distance in configuration space in order to take advantage of the fast exploratory properties of the Voronoi bias [22] offered by RRG. This biases expansion toward unexplored configuration space regions to find alternative ways of reaching 3D points.

For roadmap refinement, motion planners often use Euclidean configuration space distance, but we found that this metric can result in unintuitive tip motion. For instance, attempting to move the robot's tip a small amount may result in a motion plan where the tip takes a large, sweeping path to the goal. This is because a small tip movement can sometimes require a large change in the robot's configuration. Even if there exists a motion to directly move the robot tip to the nearby goal, that motion may have not been considered in roadmap refinement because it was not considered to connect two "near" configurations under weighted Euclidean distance. To address this issue, we define nearness for roadmap refinement to be the distance between *the 3D tip positions of each configuration*, or $d_{\text{tip}}(\mathbf{q}_1, \mathbf{q}_2) = \|\mathbf{x}(\mathbf{q}_1, 1) - \mathbf{x}(\mathbf{q}_2, 1)\|$.

2) Anatomy-based Collision Detection: When adding a configuration or edge to the roadmap, the planner must check that configurations are collision-free. This requires evaluating whether the shape of the robot $\mathbf{x}$ is in collision with an obstacle in the environment. We use a mechanically accurate kinematic model [6] of the robot to compute $\mathbf{x}$. We define the anatomical obstacles that the robot should avoid as a 3D polygonal mesh, which can be generated via segmentation from a patient's preoperative medical imaging [8]. We evaluate `isCollisionFree(`$\mathbf{q}$`)` using PQP [23], a fast collision detection algorithm that enables us to check for intersections between the anatomy meshes and a 3D mesh we quickly generate on-the-fly of the robot shape $\mathbf{x}$. We check edges $(\mathbf{q}_1, \mathbf{q}_2)$ for collisions by interpolating from $\mathbf{q}_1$ to $\mathbf{q}_2$ and collision-checking configurations along the way in `isCollisionFree(`$\mathbf{q_1}, \mathbf{q_2}$`)`.

3) Caching Shape Computations: In order to speed up offline roadmap computation and online position control, we store additional information about previous robot shape computations in each node of the roadmap. Computing the robot's shape requires solving for the initial conditions of a boundary value problem [6]. We store the solved initial conditions for each configuration in the roadmap so that they can be used as initial guesses for future shape computations of nearby configurations; this resulted in a 2.5x average speedup for shape computations.

4) Enforcing Constraints Due to Robot Design: We must ensure that our roadmap only includes feasible robot configurations, i.e., configurations must satisfy the constraints in Eq. 1. In order to sample only configurations that satisfy the constraints, we use rejection sampling. We continually sample from a set of box constraints that tightly contains the true set of valid configurations and reject those samples that violate the constraints of Eq. 1. Depending on the design of the robot, as many as 20,000 samples can be rejected before finding a valid configuration. Although this seems inefficient,

the rejection sampling procedure takes orders of magnitude less time than a single kinematic model computation.

We define the motion along an edge of the roadmap to be a linear interpolation between the two configurations. This linear interpolation works well with the constraints defined in Eq. 1 because they form a convex set, which means that if two given configurations $\mathbf{q}_1$ and $\mathbf{q}_2$ lie within the constraints, then all configurations along the linear interpolation between $\mathbf{q}_1$ and $\mathbf{q}_2$ also fulfill the constraints.

B. Interactive Planning Phase

The inputs to the interactive planning phase are the robot's kinematic model, the precomputed roadmap with associated cached data structures, and anatomical obstacle meshes. During the interactive planning phase the user continuously specifies new desired tip positions. The objective of our method is to compute robot motion plans that enable the robot's tip to follow the specified motion of the user.

The interactive planning phase operates in a loop, as shown in Fig. 2. The system first obtains the new desired tip position $\mathbf{x}_{\text{goal}}$ from the 3D mouse controlled by the user. The system then computes a motion plan from its current configuration to a new configuration such that the tip reaches $\mathbf{x}_{\text{goal}}$. The motion plan is then sent to the robot for execution, and the cycle repeats with a new tip position sent by the user. The motion planner must be fast enough for the robot to immediately begin plan execution upon the physician's selection of a new desired goal position.

1) Computing a Motion Plan: We illustrate our combined roadmap planning and IK algorithm in Alg. 1. When a new tip goal position is sent to the planner, the robot's current configuration may not lie exactly on the roadmap (we will explain why in a moment). Therefore, the algorithm first decides whether to "step back into" the precomputed roadmap by checking whether there is a configuration in the roadmap closer to the goal than the robot's current configuration. To do this we find the configuration in the roadmap nearest to the goal using the `nearestTip` routine, which uses a highly efficient nearest neighbor search [24] as implemented in the Open Motion Planning Library (OMPL) [25]. If progress toward the goal can be made by following the roadmap, we use an A^* graph search [26] to find the shortest motion plan on the roadmap to the node nearest the goal. We use d_{tip} from Sec. IV-A.1 as our cost function and heuristic to A^* to encourage plans with intuitive tip motion.

In order to reach the goal with sufficient clinical accuracy, we likely need to "step out" of the precomputed roadmap. We guide our step using the `DLS_IK` routine which implements the damped least squares (DLS) IK algorithm [27], [28]. With some problem-specific hand-tuning of the DLS parameters, DLS typically converges to sufficient accuracy within 5 iterations. When stepping into and out of the roadmap, we always perform an online collision check to ensure that the robot avoids contact with anatomical obstacles throughout the procedure.

Data: Preprocessed roadmap G with vertices and edges (V, E), anatomical obstacle meshes, current robot state $\mathbf{q}_1$ with tip position $\mathbf{x}_1$, and new desired tip position $\mathbf{x}_{\text{goal}}$

Result: Sequence of states Π describing collision-free motion from $\mathbf{q}_1$ to a state with tip position as close as possible to $\mathbf{x}_{\text{goal}}$

$\Pi \leftarrow (\mathbf{q}_1)$;
$\mathbf{q}_r \leftarrow$ nearestTip($\mathbf{x}_{\text{goal}}$, V);
$\mathbf{x}_r \leftarrow$ getTipPos($\mathbf{q}_r$);
if $\|\mathbf{x}_r - \mathbf{x}_{\text{goal}}\| < \|\mathbf{x}_1 - \mathbf{x}_{\text{goal}}\|$ **then**
    $\mathbf{q}_{\text{near}} \leftarrow$ nearestTip($\mathbf{x}_1$, V);
    if isCollisionFree($\mathbf{q}_1$, $\mathbf{q}_{\text{near}}$) **then**
        $\Pi_r \leftarrow$ AstarPath(G, $\mathbf{q}_{\text{near}}$, $\mathbf{q}_r$);
        append(Π, Π_r);

$\mathbf{q}_{\text{IK}} \leftarrow$ DLS_IK(lastElement(Π), $\mathbf{x}_{\text{goal}}$);
append(Π, $\mathbf{q}_{\text{IK}}$);
return Π

Algorithm 1: Our combined roadmap and position control planning algorithm in the interactive planning phase
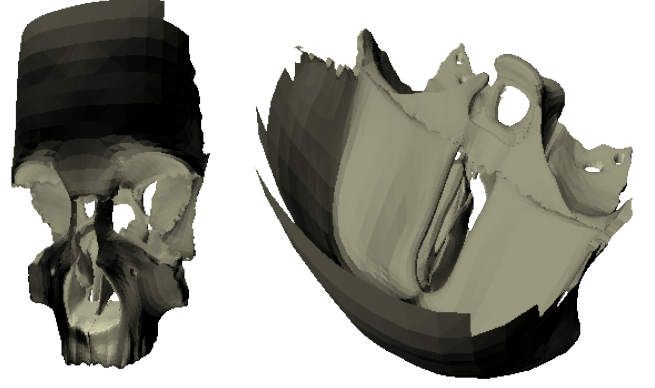

Fig. 3. 3D model of the nasal cavity and skull base used for collision detection during generation of the roadmap of collision-free concentric tube robot motion plans. Frontal view is on the left and view from above is on the right.

V. EVALUATION

We now demonstrate an example usage of our interactive motion planner on a simulated neurosurgical scenario and report results of some performance benchmarks. All computation was performed on a 2.4 GHz Intel®Xeon Quad-Core PC with 12 GB RAM.

A. Neurosurgical Scenario

Skull base tumors make up 15-20% of all primary brain tumors [29]. An endonasal approach to the skull base can save the physician from cutting healthy brain tissue, but many areas of the skull base are challenging to reach with currently available surgical devices. Concentric tube robots can curve around obstacles to navigate hard-to-reach anatomical cavities, potentially enabling treatment for previously inoperable patients.

To evaluate our interactive motion planner, we created a scenario that involves navigating a concentric tube robot through the anatomy encountered during an endonasal procedure. We used a 3D mesh model of the human nasal cavity and skull base (see Fig. 3), which we modified to reflect real-world neurosurgical conditions by removing part of the sphenoid bone to allow access to the sella. We used this mesh to generate the roadmap of collision-free plans as described in Sec. IV-A.

We implemented an interactive simulation of the neurosurgical scenario that visualizes a concentric tube robot navigating the nasal cavity model under the direct control of the user. The user can control the tip of the simulated robot by using a SensAble Phantom Desktop [4] to continually specify desired 3D tip positions for the robot to reach. Our implementation iteratively reads the position of the 3D mouse, executes the interactive planner, and visualizes the robot moving to the new tip position. The planner executes fast enough to provide intuitive, user-directed tip motion while keeping the robot safely in the bounds of the anatomical workspace. We provide snapshots of a neurosurgical simulation session in Fig. 1 and Fig. 4.

B. Experimental Evaluation

To quantitatively evaluate the performance of our new interactive motion planning method, we compared it against two related methods that each lack a key component of our combined method:

- *Roadmap only*: Use a precomputed roadmap to guide the robot toward each new goal, but do not use iterative IK to step off the roadmap for additional accuracy.
- *IK only*: Use only iterative IK to move the robot's tip to each goal. If an iterative IK step will cause a collision, motion stops at the last safe configuration.

For the roadmap-based methods, we precomputed a roadmap for the neurosurgical scenario based on 20,000 configuration samples, which resulted in 11,019 collision-free configurations and 412,056 collision-free edges.

We used each method to solve a large number of randomly generated planning queries in the neurosurgical scenario. We generated 50,000 queries by sampling pairs of 3D points $(\mathbf{s}, \mathbf{t})$ in the anatomical environment. Each point $\mathbf{s}$ defined a starting robot configuration $\mathbf{q}_1$ as the configuration in the roadmap with the nearest tip position to $\mathbf{s}$; each point $\mathbf{t}$ defined the query's goal position. For each query we executed each planner and collected computation times and goal accuracy. We only considered queries in which at least one of the three methods found a collision-free motion plan within an acceptable tolerance of the goal. We present the results of our planner comparison in Figs. 5 and 6.

Our new interactive planning method achieved the lowest tip error, with an average error of 0.0359mm. The roadmap-only method averaged a higher error of 2.03mm, which stretches the boundaries of tool precision required in minimally invasive surgery. The IK-only planner performed much worse due to its lack of roadmap routing, only safely approaching the goal within an error averaging 25.6mm.
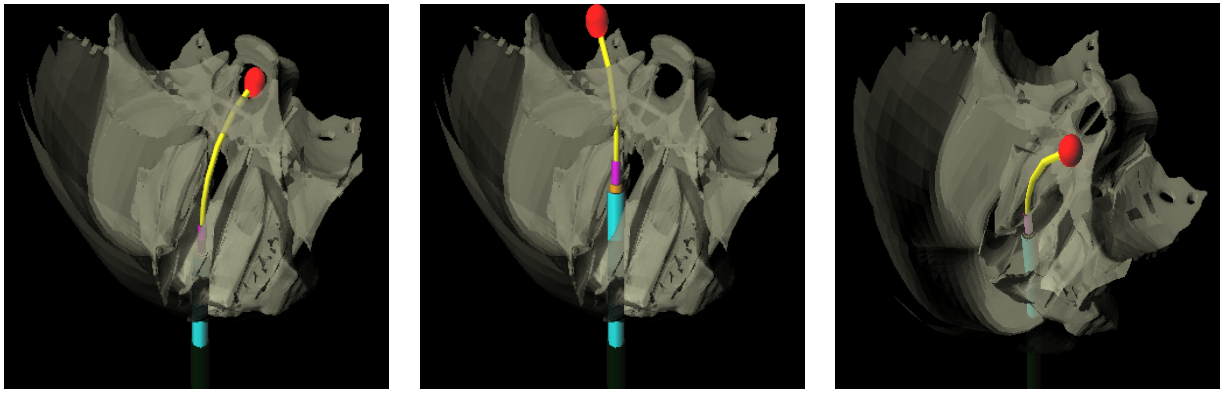
Fig. 4. Illustrations of 3 snapshots of an interactive neurosurgical simulation session. The user moves a 3D cursor (in red) using a SensAble Phantom Desktop [4] and the simulated concentric tube robot follows the cursor with its tip while avoiding contact with the rendered anatomical environment.
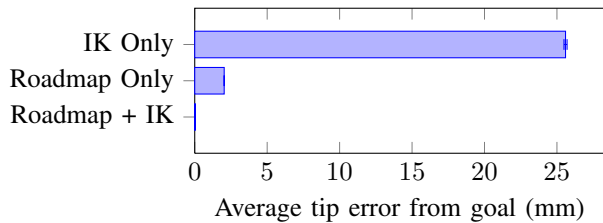

Fig. 5. Average tip error from given goals for three methods: one that combines a precomputed roadmap with iterative IK for high accuracy, one that uses only a roadmap, and one that only uses IK for position control. Our combined method, roadmap plus IK, yielded the highest average accuracy (within 0.04mm). The roadmap-only method is limited by the coarseness of the precomputed roadmap, and the pure position control method is highly hindered by its lack of obstacle avoidance. 95% confidence intervals are shown.
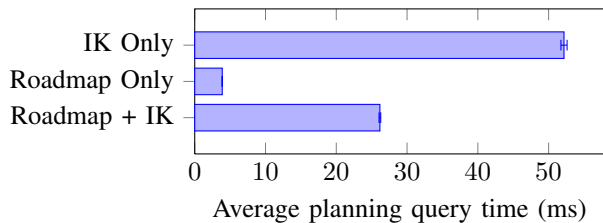

Fig. 6. Average planning query execution times of three versions of our methods: one that combines a precomputed roadmap with iterative IK, one that uses only a roadmap, and one that only uses IK for position control. Using a precomputed roadmap helps enable execution of queries at interactive rates.

Our interactive motion planner spent an average of 26.2ms on each query for an average query rate of 38Hz, which is fast enough for intuitive interaction. We also note that the precise motions required in a real surgical procedure will likely yield queries over shorter distances than our random queries, likely resulting in even higher query rates. The roadmap-only method yielded a faster average query time than the interactive motion planner because it does not perform the robot shape computations and collision detection required for the safe and accurate position control. We note that the roadmap-only method's speedup comes at the expense of larger tip error from goals (see Fig. 5).
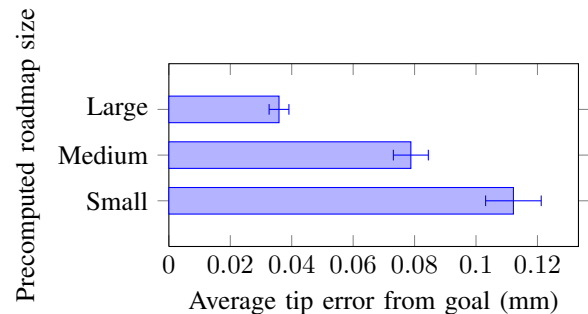

Fig. 7. Average tip error from given goal of our new planning method, using precomputed roadmaps of three different sizes.

C. Effect of Roadmap Size on Planner Performance

We also investigated the effects of varying the size of the precomputed roadmap used in our new interactive planner. We computed three roadmaps (small, medium, and large) by generating 5,000, 10,000, and 20,000 configuration samples. The final roadmaps had 3,113, 5,837, and 11,019 collision-free configurations, respectively. These roadmaps occupied 110MB, 166MB, and 350MB of RAM, respectively. The roadmaps required 1 hour, 3 hours, and 6 hours, respectively, to generate. We note that all these computation times lie within the typical timeframe between preoperative medical imaging and the actual surgery. We then executed our interactive motion planner on the same set of planning queries from Sec. V-B using each of these roadmaps. We show the results in Figs. 7 and 8.

Although generating larger roadmaps requires more computation time, our experiments show that larger roadmaps result in both faster planning queries and lower tip error. With a larger roadmap, the roadmap-based planner is able to find motion plans to configurations nearer to the given goal, thereby reducing the expensive iterative IK computations necessary to move the tip exactly to the goal.

VI. CONCLUSION

We present a motion planning method that can compute collision-free plans for concentric tube robots at interactive
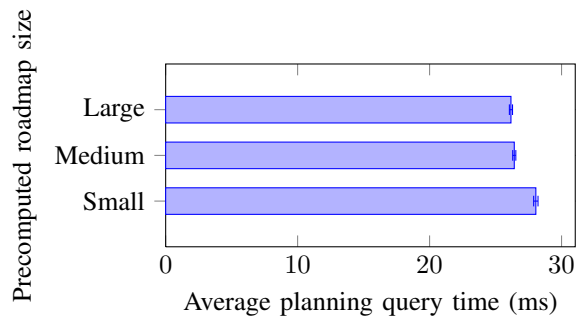
Fig. 8. Average planning query execution times of our new planning method, using precomputed roadmaps of three different sizes.

rates. Our planner's speed enables users to continuously move the robot's tip while the planner ensures obstacle avoidance for the robot's shaft. Our method derives its speed and accuracy by combining offline precomputation of a collision-free roadmap with online position control.

We envision this interactive motion planner as a component of a larger teleoperative system for concentric tube robots. In future work we will use our new planner to control a physical robot in a phantom anatomical environment. This system's user interface can be improved by using recent work in shared teleoperation [30], [31]. We will explore uncertainty models to make the planner more robust to errors in the kinematic and anatomical modeling. We will also investigate extending our planner to consider dynamic anatomical obstacles like breathing lungs and beating hearts, which will require fast online repair of the precomputed roadmap according to intraoperative medical imaging.

VII. Acknowledgement

References

[1] J. Burgner, P. J. Swaney, D. C. Rucker, H. B. Gilbert, S. T. Nill, P. T. Russell III, K. D. Weaver, and R. J. Webster III, "A bimanual teleoperated system for endonasal skull base surgery," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Sep. 2011, pp. 2517–2523.

[2] L. G. Torres, R. J. Webster III, and R. Alterovitz, "Task-oriented design of concentric tube robots using mechanics-based models," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 4449–4455.

[3] A. H. Gosline, N. V. Vasilyev, A. Veeramani, M. Wu, G. Schmitz, R. Chen, V. Arabagi, P. J. del Nido, and P. E. Dupont, "Metal MEMS tools for beating-heart tissue removal," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, St. Paul, May 2012, pp. 1921–1926.

[4] "Sensable Phantom Desktop Overview." [Online]. Available: http://geomagic.com/en/products/phantom-desktop/overview

[5] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Trans. Robotics*, vol. 26, no. 2, pp. 209–225, Apr. 2010.

[6] D. C. Rucker, "The mechanics of continuum robots: model-based sensing and control," Ph.D. dissertation, Vanderbilt University, 2011.

[7] R. Xu, A. Asadian, A. S. Naidu, and R. V. Patel, "Position control of concentric-tube continuum robots using a modified Jacobian-based approach," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2013, pp. 5793–5798.

[8] H. J. Johnson, M. McCormick, L. Ibáñez, and Insight Software Consortium, "The ITK Software Guide," Available: http://www.itk.org/ItkSoftwareGuide.pdf, Dec. 2013.

[9] P. E. Dupont, J. Lock, and E. Butler, "Torsional kinematic model for concentric tube robots," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2009, pp. 3851–3858.

[10] D. C. Rucker and R. J. Webster III, "Parsimonious evaluation of concentric-tube continuum robot equilibrium conformation," *IEEE Trans. Biomedical Engineering*, vol. 56, no. 9, pp. 2308–2311, Sep. 2009.

[11] P. Sears and P. E. Dupont, "A steerable needle technology using curved concentric tubes," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2006, pp. 2850–2856.

[12] D. C. Rucker, B. A. Jones, and R. J. Webster III, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE Trans. Robotics*, vol. 26, no. 5, pp. 769–780, Jan. 2010.

[13] J. Lock, G. Laing, M. Mahvash, and P. E. Dupont, "Quasistatic modeling of concentric tube robots with external loads," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2010, pp. 2325–2332.

[14] L. A. Lyons, R. J. Webster III, and R. Alterovitz, "Motion planning for active cannulas," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2009, pp. 801–806.

[15] K. Trovato and A. Popovic, "Collision-free 6D non-holonomic planning for nested cannulas," in *Proc. SPIE Medical Imaging*, vol. 7261, 2009.

[16] L. G. Torres and R. Alterovitz, "Motion planning for concentric tube robots using mechanics-based models," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Sep. 2011, pp. 5153–5159.

[17] J. Bruce and M. Veloso, "Real-time randomized path planning for robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, EPFL, Lausanne, Switzerland, Oct. 2002, pp. 2383–2388.

[18] M. Zucker, J. Kuffner, and M. Branicky, "Multipartite RRTs for rapid replanning in dynamic environments," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Roma, Italy, Apr. 2007, pp. 1603–1609.

[19] C. Stachniss and W. Burgard, "An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, vol. 1, EPFL, Lausanne, Switzerland, Oct. 2002, pp. 508–513.

[20] J. P. van den Berg, "Anytime path planning and replanning in dynamic environments," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006, pp. 2366–2371.

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

[22] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[23] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast proximity queries with swept sphere volumes," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, San Francisco, CA, Apr. 2000, pp. 3719–3726.

[24] S. Brin, "Near neighbor search in large metric spaces," in *Proc. 21st Conf. on Very Large Databases (VLDB)*, Zurich, Switzerland, 1995, pp. 574–584.

[25] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics and Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec. 2012. [Online]. Available: http://ompl.kavrakilab.org

[26] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[27] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *J. Dynamic Systems, Measurement, and Control*, vol. 108, pp. 163–171, 1986.

[28] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. Systems, Man and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.

[29] "American brain tumor association (ABTA)." [Online]. Available: http://abta.org

[30] K. Hauser, "Recognition, prediction, and planning for assisted teleoperation of freeform tasks," in *Proc. Robotics: Science and Systems*, Sydney, Australia, Jul. 2012.

[31] A. D. Dragan and S. S. Srinivasa, "Formalizing assistive teleoperation," in *Proc. Robotics: Science and Systems*, Sydney, Australia, Jul. 2012.