Interactive 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 taking curved paths to difficult-to-reach areas of a patient's anatomy. Operating these devices is hard due to their complex and unintuitive kinematics. Existing position control methods for driving the robot's tip can cause the robot to collide with sensitive structures in the anatomy associated with patient injury, and prior motion planning methods that offer collision avoidance are too slow for a surgeon to interactively control the robot during a procedure. In this paper, we present a motion planning method that can compute collision-free motion plans for concentric tube robots at interactive rates. Our planner's high speed enables surgeons to continuously and freely drive the robot's tip while the planner ensures that the robot's shaft does not collide with any anatomical obstacles. Our method 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 miniaturized tentacle-like robotic devices designed for minimally invasive surgery. Their curving ability and small size allow them to reach anatomical sites previously inaccessible by currently available surgical instruments. Concentric tube robots may therefore enable new, safer surgical access to a variety of sites in the human body, including the skull base [1], the lungs [2], [3], and the heart [4].

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. However, this powerful shape-changing property also poses a major challenge: their complex and unintuitive kinematics. The overall shape of a concentric tube robot is determined by deep mechanical interactions between the device's curved elastic tubes. A surgeon 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 therefore look to computation to tame the complexity of these devices and allow for intuitive guidance by physician input. Kinematic modeling of concentric tube robots has made great strides in recent years, allowing for enough speed and accuracy in shape computation to achieve interactive position control of the robot's tip [6], [7], [8]. However, these methods do not account for obstacles. Collisions with anatomical obstacles can increase risk to the patient and can bend the device in an unpredictable manner that impedes 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 surgeon successfully steers the tip clear of obstacles, the entire shape of the robot shaft can sometimes change dramatically to accommodate a given tip position, which can result in a tube's shaft colliding with anatomical obstacles.

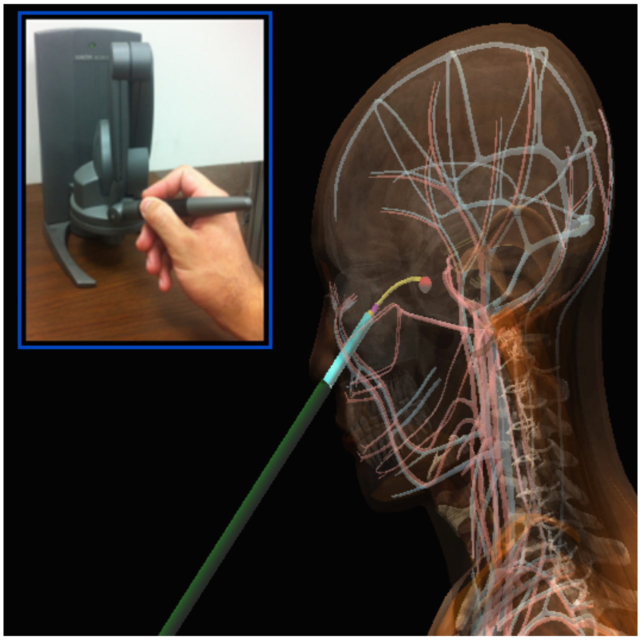

Fig. 1. Virtual simulation of a concentric tube robot being controlled with a SensAble Phantom Desktop [5]. Our interactive motion planner enables the user to move the robot's tip while ensuring that the robot's entire shaft avoids contact with known sensitive anatomical structures.

In this paper, we present a motion planning method that can computes collision-free motion plans for concentric tube robots at interactive rates. This has the potential to allow the physician to continuously specify a desired tip location for the concentric tube robot using a 3D mouse (e.g., a SensAble Phantom [5]), and the robot can continuously respond by reshaping the concentric tube robot to reach the desired tip position while ensuring that the entire device shaft avoids anatomical obstacles.

Our method assumes a pre-operative image, such as a CT scan or MRI, is obtained prior to the procedure, as is typical for complex surgical procedures. From these images relevant structures can be segmented either manually or using automatic segmentation software [9]. Given the segmented

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

anatomical obstacles, prior work has investigated motion planning for concentric tube robots to compute sequences of tube rotations and extensions to automatically reach goals while avoiding obstacles [10], [11], [12]. However, the high computational cost of accurately evaluating the robot's kinematics (which requires solving a numerical system for clinically-acceptable accuracy [13], [14]) means that the collision detection steps of previous motion planners are too slow for the planners to be interactively used by a surgeon during a procedure. Sacrificing accurate kinematic modeling for faster computation is not an option either, because excess error in the kinematic model can lead to the robot unexpectedly colliding with sensitive anatomical structures and risking injury to the patient.

In this paper we achieve interactive rates by creating a motion planner specifically designed for conecntric 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. This way, we can leave most of the expensive robot shape computations to an offline step, saving a huge amount of computation during the actual surgical procedure. For roadmap construction, we use a configuration sampling method designed for fast exploration of the constrained spaces of the human body, then connect the configurations in such a way that allows for smooth, intuitive tip motion during operation. We also cache information about the shape computations performed during roadmap construction to enable fast online path queries. We then combine this roadmap-based planning approach with a position control method using iterative inverse kinematics to reach user-specified positions not exactly represented in the precomputed roadmap. This results in a method that quickly computes a collision-free motion plan to the region of interest, and then uses fast position control to locally guide the robot tip to exactly the position required by the surgeon.

We demonstrate our new 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 [15], to torsionally compliant models [14], [13], to models that consider external loading [16], [17]. In this paper we use a highly accurate model developed by Rucker et al. [7].

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) inverse kinematics [7], [8]. These position control methods do not take obstacles into account, and can fail to converge on the required goal position. However, we do use position control in order to increase the accuracy of our collision-free motion planner. 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 inverse kinematics [6]. This approach feasibly allows for obstacle avoidance, but position control does not benefit from the robot's redundant degrees of freedom to consider alternate configurations for reaching goals.

In order to compute collision-free motion plans for mechanically accurate models of concentric tube robots, Torres et al. used rapidly-exploring roadmaps (RRM) combined with Jacobian-based goal biasing [11]. However, the computation time required for accurate robot shape computation makes online generation of an RRM too slow for interactive planning. Trovato and Popovic assumed a simplified, less accurate kinematic model and designed the tube curvatures to allow for a collision-free path to a clinical site [12].

Lots of prior work has focused on speeding up collision-free motion planning algorithms for various problems. Many of these approaches have been developed for motion planning in dynamic environments, where motion plans must be computed quickly enough to remain valid under movement of obstacles. Some work includes re-use of paths from prior planning iterations [19], [20], lower-dimensional grid searches to guide the full-dimensional search [22], and repairing paths precomputed in a static environment [23]. In this paper, we assume static anatomical obstacles, but we are hampered by kinematics computations that are too expensive for the online planning and collision detection required by most of the previous replanning methods. Our method is most similar to that of van den Berg et al. [23] 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 to step out of the precomputed roadmap for additional accuracy.

Much work has also addressed collision-free roadmap generation. The classical approach is a sampling-based motion planner called the probabilistic roadmap (PRM) [24]. The default configuration sampling used in PRM, however, does not work as well with the highly constrained anatomical environments that concentric tube robots are designed to navigate, where large portions of the sampleable configuration space may not be reachable from a given starting point. Growing a roadmap from a given starting state is captured in the rapidly-exploring random graph (RRG) [25]. Our roadmap generation method is based on RRG, with a modification to use different distance metrics for roadmap expansion and roadmap refinement.

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 in isolation 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 $\mathbf{v}_{\text{start}}$.

Each tube may be (1) inserted or retracted from the previous tube, and (2) axially rotated, yielding 2 degrees of freedom per tube. We therefore define a concentric tube robot configuration as 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 $\mathcal{Q}$ of all concentric tube robot configurations is $\mathcal{Q} = (S^1)^N \times \mathbb{R}^N$.

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 physical design of a concentric tube robot. 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 [7] requires that the inserted length of tubes increase with decreasing cross-sectional radius. A set of tube insertion values $\boldsymbol{\beta}$ is therefore considered valid only if the following equations hold (for $N \geq 2$):

$$\begin{aligned} & -l_{\text{track}} \leq \beta_1 \\ & \beta_{i-1} + \delta \leq \beta_i \quad \text{for } i = 2, \dots, N \\ & \beta_{i-1} + l_s^{i-1} + l_c^{i-1} \geq \beta_i + l_s^i + l_c^i \quad \text{for } i = 2, \dots, N \\ & \beta_N < 0 \\ & \beta_N + l_s^i + l_c^i \geq 0 \end{aligned} \tag{1}$$

For a given configuration $\mathbf{q}$, 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 the domain $[0, 1]$. We note $\mathbf{x}(\mathbf{q}, 0)$ maps to $\mathbf{x}_{\text{start}}$ 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. [7].

B. Motion Planning Problem Formulation

We define a path Π in the robot's configuration space as a sequence of configurations $(\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n)$. A collision-free path is a path where, for $i = 1, \dots, n$, the shape of the robot corresponding to $\mathbf{q}_i$ does not collide with any obstacles in the environment.

Anatomical obstacles can be defined in any geometric representation, as long as it is possible to compute a predicate function `is_collision_free`($\mathbf{q}$) 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 general 3D polygonal meshes, which can be generated from medical image segmentations [9].

We aim for the robot tip to reach a goal point $\mathbf{x}_{\text{goal}}$. We assume that the start location $\mathbf{x}_{\text{start}}$ and initial orientation $\mathbf{v}_{\text{start}}$ are fixed and correspond to the orifice through which the device is deployed based on the clinical procedure being performed.

Given the concentric tube robot's properties, the start configuration $\mathbf{q}_1$, the goal coordinate $\mathbf{x}_{\text{goal}}$, and a representation of anatomical obstacles, we formulate the motion planning problem as a search for a path $\Pi = (\mathbf{q}_1, \dots, \mathbf{q}_n)$ where:

1) $\|\mathbf{x}(\mathbf{q}_n, 1) - \mathbf{x}_{\text{goal}}\|$ is minimized;
2) each configuration along Π is collision-free;
3) the path results in smooth tip motion.

Additionally, we wish to compute these collision-free paths quickly enough to enable interactive control the concentric tube robot's tip.

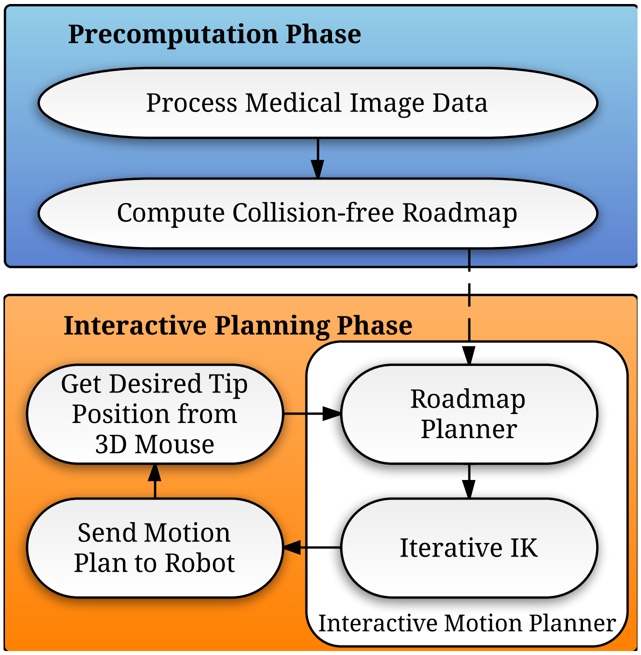
Precomputation Phase
Process Medical Image Data
Compute Collision-free Roadmap
Interactive Planning Phase
Get Desired Tip Position from 3D Mouse
Roadmap Planner
Send Motion Plan to Robot
Iterative IK
Interactive Motion Planner

Fig. 2. Method overview.

IV. Accurate and Interactive Motion Planning

In order to achieve interactive collision-free motion planning for a concentric tube robot, our approach blends sampling-based roadmap planning (which considers global routing through the robot's free configuration space) with a fast, iterative IK solver (which solves for local adjustments for high accuracy).

Our method proceeds in two phases: a *precomputation phase* followed by an *interactive planning phase*, as shown in Figure 2. In the precomputation phase, we generate a roadmap using a variant of the rapidly-exploring random graph (RRG), which we describe in Sec. IV-A. We cache shape 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 path from the robot's current configuration to a desired tip position using a 3D mouse. The precomputed roadmap provides collision-free paths to reachable areas in the anatomy, and the online iterative IK allows fine-grained control of the robot to specific points in these areas.

A. Precomption Phase

The input to the precomputation phase is a representation of the anatomy (for collision detection), the design parameters of the concentric tube robot (to accurately compute kinematics), and the start location and direction vector of the robot. The output of the precomputation phase will be 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 where each edge encodes a collision-free motion between the configurations of the corresponding nodes. We compute this roadmap using RRG [25] with some customizations for our application.

The RRG algorithm builds a roadmap by beginning at a given starting configuration, and only adding configuration samples to the graph that can be connected by a collision-free motion to a configuration already in the roadmap. RRG also refines the roadmap by including adding edges between configurations in such a way that guarantees *asymptotic optimality* of path quality under certain assumptions. The result of roadmap generation is a graph that can quickly be queried for high-quality collision-free paths between any two configurations in the roadmap. Our implementation of RRG differs in the strategy used for refining the roadmap.

We execute our customized RRG algorithm for some number of iterations in order to generate our collision-free roadmap. We then save the roadmap to the hard disk in a format that can be quickly reloaded for use during the surgical procedure. In the following paragraphs we describe details of the RRG customizations that enable interactive 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 two 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 [27] offered by RRG. This means that we bias toward expansion of unexplored regions of the configuration space in order to find many alternative ways of reaching 3D points.

For roadmap refinement, motion planners often use Euclidean configuration space distance, but we found that this metric makes interactive control of the robot tip unintuitive. For instance, attempting to move the robot's tip a small amount may result in a motion plan in which the tip takes a large, sweeping trajectory to the goal. This results from the fact that, in many cases, moving the robot's tip a small amount requires a large change in the robot's configuration. Even if there exists a motion to smoothly move the robot tip to the nearby goal, that motion may have not been considered in roadmap refinement because, under weighted Euclidean distance, that motion was not considered to connect two "near" configurations. To address this issue, we define nearness for roadmap refinement to be the distance between *the 3D tip positions of each configuration*, or $d(\mathbf{q}_1, \mathbf{q}_2) = \|\mathbf{x}(\mathbf{q}_1, 1) - \mathbf{x}(\mathbf{q}_2, 1)\|$.

2) Anatomy-based Collision Detection: When adding a configuration or edge to the roadmap, the motion 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. As described in Section III-A, we use a mechanically accurate kinematic model of the robot to compute $\mathbf{x}$. We define the anatomical obstacles the robot should avoid during the clinical procedure as a 3D polygonal mesh. Such a mesh can be generated from a patient's preoperative medical imaging via manual or automatic segmentation algorithms [9]. We evaluate $\texttt{is\_collision\_free}(\mathbf{q})$ using PQP [28], 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 $\mathbf{x}$.

3) Motion Between Configurations: During roadmap construction, we need a local controller to define the motion from one configuration to another that defines an edge in the roadmap graph. We use a simple linear interpolation as the local controller. This linear interpolation works well with the constraints defined in Eq. 1; these constraints in fact 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 $\mathbf{q_2}$ will also fulfill the constraints.

4) 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 (BVP) [7]. 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 can result in 10x speedups for shape computations.

5) Enforcing Constraints Due to Robot Design: To ensure that our roadmap only includes feasible robot configurations, we need to ensure that all configurations considered fulfill 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 contain the true set of valid configurations, rejecting those samples that fall outside of the constrained set. Depending on the design of the robot's tubes, as many as 20,000 samples can be rejected before finding a valid configuration. Although this seems inefficient, the total time spent in the rejection sampling procedure is multiple orders of magnitude less than the time spent on a single kinematic model computation.

B. Interactive Planning Phase

The input to the interactive planning phase is the precomputed roadmap with associated cached data structures as well as the robot's kinematic model. During the interactive planning phase the user continuously specifies the desired tip position. 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.

1) Continuous Replanning Loop: The interactive planning phase operates in a loop, as shown in Figure 2. The system first obtains the desired tip position $\mathbf{p}_t$ 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 $\mathbf{p}_t$. The motion plan is computed using the precomputed roadmap combined with iterative IK position control. The motion plan is then sent to the robot for execution, and the cycle repeats. For the continuous replanning approach to work intuitively for the user, the motion planner must be fast to prevent perceived lag.

2) Computing a Motion Plan: We illustrate our combined roadmap planning and IK algorithm for interactive planning in Alg. 1. Given a tip goal position, 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 [29] as implemented in the Open Motion Planning Library (OMPL) [30]. If progress toward the goal can be made by following the roadmap, we use an A^* graph search [31] to find the shortest path on the roadmap to the node nearest the goal. We use the distance between the tip positions of each configuration as our cost function and heuristic to A^* to encourage plans with smooth tip motion.

In order to exactly reach the goal with sufficient clinical accuracy, we likely need to "step out" of the precomputed roadmap. We guide our step using the `DLSIK` routine which implements the damped least squares (DLS) IK algorithm [32], [33]. With some hand-tuning of the DLS parameters specifically for our problem, our method typically converges to sufficient accuracy in about 5 iterations of DLS.

When stepping off and onto 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 vertex and edge
      sets $(V, E)$, current robot state $\mathbf{q}_0$ with tip
      position $\mathbf{p}_0$, and desired tip position $\mathbf{p}_t$
Result: Sequence of states $\Pi$ describing collision-free
        motion from $\mathbf{q}_0$ to a state with tip position as
        close as possible to $\mathbf{p}_t$
$\mathbf{q}_{\text{current}} \leftarrow \mathbf{q}_0$;
$\Pi \leftarrow (\mathbf{q}_0)$;
$\mathbf{q}_r \leftarrow$ nearestTip($\mathbf{p}_t$, $V$);
$\mathbf{p}_r \leftarrow$ getTipPos($\mathbf{q}_r$);
if $\|\mathbf{p}_r - \mathbf{p}_t\| < \|\mathbf{p}_0 - \mathbf{p}_t\|$ then
    $\mathbf{q}_n \leftarrow$ nearestTip($\mathbf{p}_0$, $V$);
    $\Pi_r \leftarrow$ AstarPath($G$, $\mathbf{q}_n$, $\mathbf{q}_r$);
    append($\Pi$, $\Pi_r$);

$\mathbf{q}_{\text{IK}} \leftarrow$ DLS_IK ($\mathbf{q}_{\text{current}}$, $\mathbf{p}_t$);
append($\Pi$, $\mathbf{q}_{IK}$);
return $\Pi$
```

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

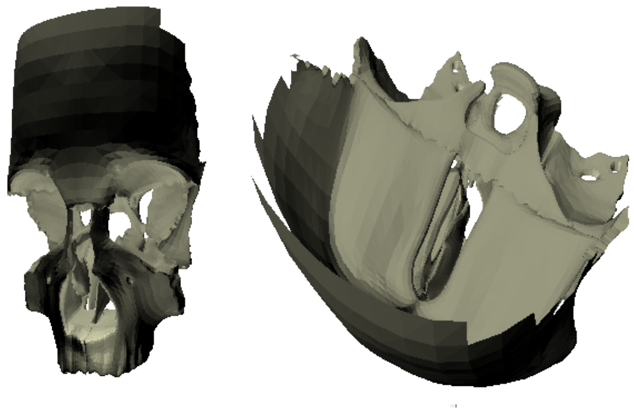

Fig. 3. 3D model of the nasal cavity and skull base used for collision detection during generation of the roadmap of collision-free concentric tube robot motion plans. Frontal view is on the left, and view from above is on the right.

V. Evaluation

In this section we demonstrate an example usage of our interactive motion planner on a simulated neurosurgical scenario, as well as results of some performance benchmarks. All computation was performed on a 2.40 GHz Intel®Xeon Quad-Core PC with 12 GB RAM.

A. Neurosurgical Scenario

Tumors commonly arise in the skull base, with skull base tumors making up 15-20% of all primary brain tumors [34]. An endonasal approach to the skull base through the nasal cavity can save the surgeon 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 previously inoperable patients to benefit from an endonasal surgical treatment.

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. We modified the 3D mesh model to reflect real-world neurosurgical conditions in which the surgeon removes part of the sphenoid bone to allow access to the sella, where the pituitary gland is located. We used this mesh to generate the roadmap of collision-free paths 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 from Sec. V-A under the direct control of the user. The user can control the tip of the simulated robot by using a SensAble Phantom Desktop [5] to continually specify desired 3D tip positions for the robot to reach. Our implementation iteratively executes the interactive planner with each input tip position and visualizes the concentric tube robot performing the computed motion plans. This results in the user moving a 3D cursor in the environment and the robot moving such that its tip reaches the cursor while the robot's entire shaft avoids contact with the anatomical environment. The planner executes fast enough to provide smooth, satisfying tip motion as directed by the user while remaining 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 its performance 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 the 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 the goal. We add collision detection so that, if a step of iterative IK will cause a collision, motion stops at the last safe configuration until the next tip position is provided as input.

For both the our interactive motion planner and the roadmap-only planner, we precomputed a roadmap for the neurosurgical simulation based on 20,000 configuration samples, which resulted in a roadmap that contained 11,019 collision-free configurations and 412,056 collision-free edges.

To benchmark the methods, we used each method to execute a large number of randomly generated path queries in the neurosurgical scenario. We generated 50,000 path queries, each by sampling a pair of 3D points $(\mathbf{s}, \mathbf{t})$ from a bounding box roughly corresponding to the empty space in our collision mesh. The first point $\mathbf{s}$ defined a starting robot configuration $\mathbf{q}_0$ by taking the configuration in the precomputed roadmap with the tip position $\mathbf{p}_0$ nearest to $\mathbf{s}$; the second point $\mathbf{t}$ defined the goal position of this query. For each query we executed each planner and collected query execution times and goal accuracy. Since some queries are infeasible given the robot and anatomical constraints, we only considered path queries in which at least one of the three planners found a collision-free path to a point within an acceptable tolerance of the goal. We present the results of our planner comparison in Figs. 5 and 6.

Over all feasible path queries, our new interactive planning method achieved the lowest tip error, with an average error of 0.0359mm. The roadmap-only method without additional position control 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 collision-free roadmap routing, only safely approaching the goal within an error averaging 25.6mm. Our combined planner can therefore compute collision-free paths to the goal with highest accuracy.

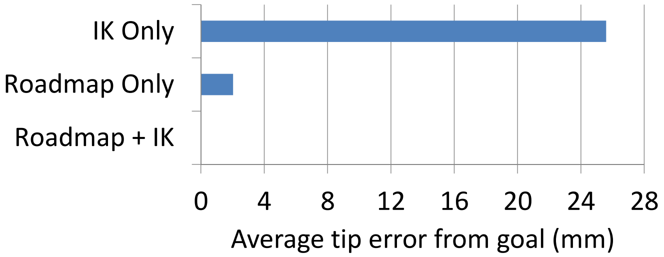

Fig. 5. Average tip error from given goal for three versions of our planning method: one that combines a precomputed roadmap with iterative IK for high accuracy, one that uses only a roadmap, and one that only uses IK for position control. Our interactive motion planner that combines both methods yielded the highest average accuracy (within 0.0001mm). The roadmap method that does not use position control is limited by the coarseness of the precomputed roadmap, and the pure position control method is highly hindered by its lack of obstacle avoidance.

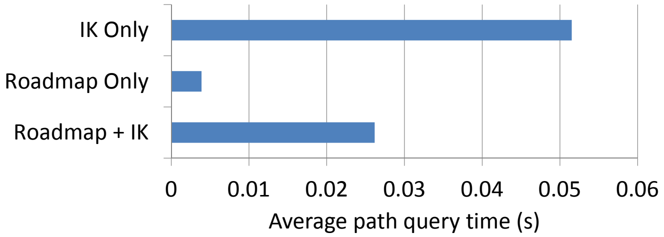

Fig. 6. Average path query execution times of three versions of our planning method: one that combines a precomputed roadmap with iterative IK for high accuracy, one that uses only a roadmap, and one that only uses IK for position control. Using a precomputed roadmap provides execution of path queries at sufficient speeds for interactive collision-free tip control.

Our interactive motion planner spent an average of 26.2ms on each path query for an average query rate of 38Hz, which is fast enough for intuitive interaction. We also note that the precise and deliberate motions required in a real surgical procedure will likely yield consecutive path queries over shorter distances than those generated for these experiments, which means the path query rate in a real procedure will likely be even higher. 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 of the interactive motion planner. We note that the roadmap-only method's speedup comes at the expense of larger tip error from goal (see Fig. 5).

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 generated three roadmaps by running the precomputation phase for 5,000, 10,000, and 20,000 configuration samples. The final roadmaps had 3,113, 5,837, and 11,019 collision-free configurations, respectively. The roadmaps required 1 hour, 3 hours, and 6 hours, respectively, to generate. We note that all these computation times lie well within the typical timeframe between preoperative medical scanning and the

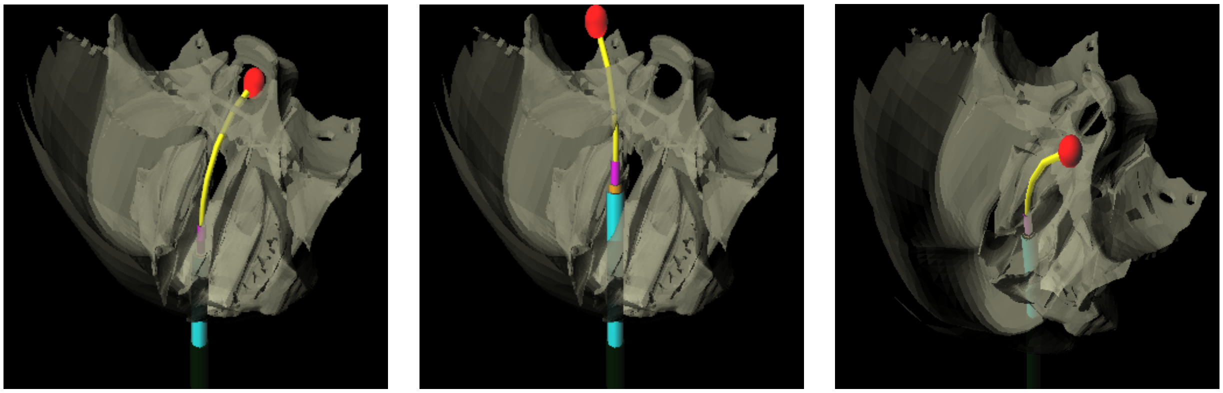

Fig. 4. Illustrations of 3 snapshots of an interactive neurosurgical simulation session. The user moves a 3D cursor (in red) using a SensAble Phantom Desktop [5] and the simulated concentric tube robot follows the cursor with its tip while avoiding contact with the rendered anatomical environment.

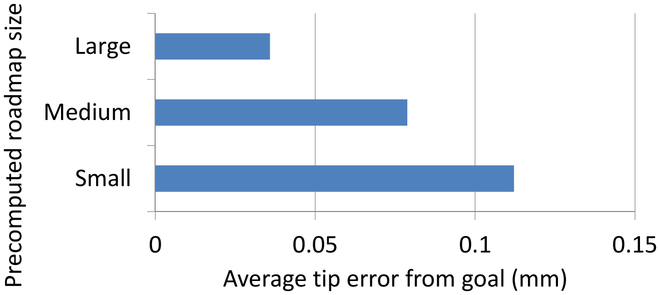

Fig. 7. Average tip error from given goal of our new planning method, using precomputed roadmaps of three different sizes.

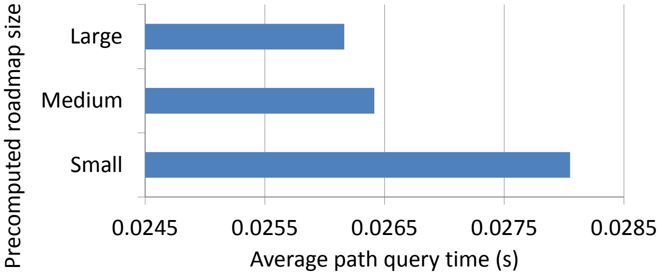

Fig. 8. Average path query execution times of our new planning method, using precomputed roadmaps of three different sizes.

actual surgery for many procedures. We then executed our interactive motion planner on the same randomly sampled set of path queries from Sec. V-B using each of these precomputed roadmaps for collision-free path finding. We show the results in Figs. 7 and 8.

Although there is an initial cost in computation time for generating larger roadmaps, our experiments show that larger roadmaps result in both faster path query computation and lower tip error. With more configuration samples in the roadmap, the roadmap-based planner is able to find paths to configurations nearer to the given goal, therby reducing the obstacle-unaware iterative IK computations necessary to drive exactly to the goal. Iterative IK position control requires expensive computation of the robot's kinematic model, so fewer iterative IK calculations mean faster queries. Interestingly, this savings in computation time even overcomes the increase in computation time associated with executing the A^* graph search on a larger graph.

VI. Conclusion

We present a motion planning method that can compute collision-free plans for concentric tube robots at interactive rates. Our planner's high speed enables users to continuously and freely drive the robot's tip while the planner ensures that the robot's shaft does not collide with any known anatomical obstacles. Our method derives its speed and accuracy by combining offline precomputation of a collision-free roadmap with online position control.

We envision this fast 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 could be made even more intuitive for surgeons by using recent work in shared teleoperation to enable user intent prediction [26], [37]. We will also investigate ways of incorporating uncertainty models to make the interactive motion planner more robust to errors in the kinematic modeling and to uncertainty in the anatomy. We will also explore ways of intuitively providing more information to the surgeon, such as applying haptic feedback as the robot approaches the boundary of the reachable workspace. 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 in order to maintain roadmap validity.

VII. Acknowledgement

This material is based upon work supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. DGE-1144081. This research is also supported in part by NSF grant #IIS-0905344 and by the National Institutes of Health (NIH) under grant #R21EB011628. We thank Hunter Gilbert from Vanderbilt University for his insights on accurate kinematic modeling of concentric tube robots.

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. A. Lyons, R. J. Webster III, and R. Alterovitz, "Planning active cannula configurations through tubular anatomy," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2010, pp. 2082–2087.

[3] 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.

[4] 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.

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

[6] 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.

[7] D. C. Rucker, "The Mechanics of Continuum Robots: Model-Based Sensing and Control," Ph.D. dissertation, Vanderbilt University, 2011.

[8] 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.

[9] L. Ibáñez, W. Schroeder, L. Ng, J. Cates, and Insight Software Consortium, "The ITK Software Guide, Second Edition, Updated for ITK version 2.4," Available: http://www.itk.org/ItkSoftwareGuide.pdf, Nov. 2005.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] J. Lock and P. E. Dupont, "Friction Modeling in Concentric Tube Robots," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 1139–1146.

[19] 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.

[20] 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.

[21] J. Vannoy and J. Xiao, "Real-time adaptive motion planning (RAMP) of mobile manipulators in dynamic environments With unforeseen changes," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1199–1212, Oct. 2008.

[22] 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.

[23] 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.

[24] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Trans. Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[25] 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.

[26] K. Hauser, "Recognition, Prediction, and Planning for Assisted Teleoperation of Freeform Tasks," in *Proc. Robotics: Science and Systems*, Sydney, NSW, Jul. 2012.

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

[28] 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.

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

[30] I. A. Sucan, 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

[31] 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.

[32] 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.

[33] 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.

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

[35] "Full male head anatomy." [Online]. Available: http://www.turbosquid.com/3d-models/max-male-head-anatomy-skeleton/704087

[36] "Object-oriented graphics rendering engine (OGRE)." [Online]. Available: http://www.ogre3d.org/

[37] A. D. Dragan and S. S. Srinivasa, "Formalizing Assistive Teleoperation," in *Proc. Robotics: Science and Systems*, Sydney, NSW, Jul. 2012.