Following Surgical Trajectories with Concentric Tube Robots via Nearest-Neighbor Graphs

Sherdil Niyaz[1], Alan Kuntz[2], Oren Salzman[3],
Ron Alterovitz[2], and Siddhartha S. Srinivasa[1]

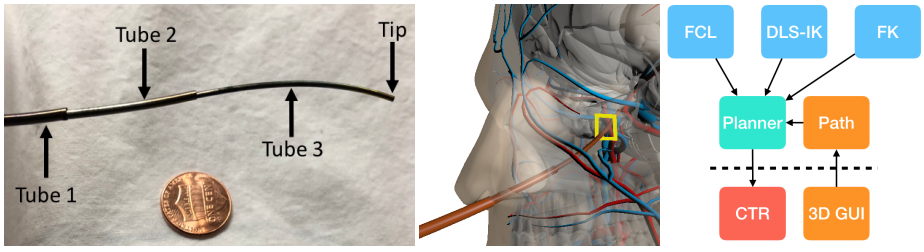
[1] School of Computer Science and Engineering, University of Washington
[2] Department of Computer Science, University of North Carolina at Chapel Hill
[3] The Robotics Institute, Carnegie Mellon University

1 Introduction

Concentric tube robots, or CTRs, are tentacle-like robots composed of precurved telescoping tubes (Fig. 1a) and are controlled by rotating and translating each individual tube [6]. Their dexterity and small diameter enable minimally-invasive surgery in constrained areas, such as accessing the pituitary gland via the sinuses. Unfortunately, their unintuitive kinematics make manually guiding the tip while also avoiding obstacles with the entire tentacle-like shape extremely difficult [19]. This motivates a need for new user interfaces and planning algorithms.

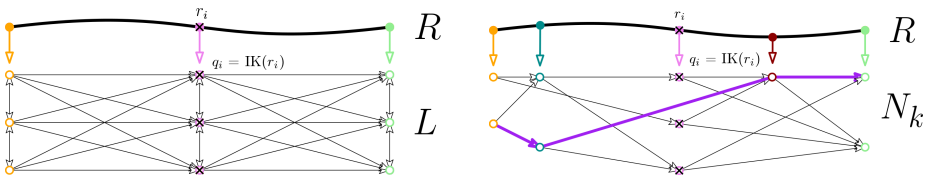
Although existing planners [19] enable CTRs to reach specified points in task-space, this is often insufficient. For example, cutting a window in the skull during brain surgery (Fig. 1b) requires specifying an entire *path* R for the robot's tip. We have, to the best of our knowledge, implemented the first planner in this domain able to compute a trajectory that closely follows such a task-space path.

Algorithms that are able to follow some task-space path R by computing a constrained path in configuration space ($\mathcal{C}$-space) [2,14,17,22] have two main challenges: (i) some (e.g., a vector-field planner [17]) are fast but myopic, often falling into local minima and (ii) all are oblivious to how closely the generated paths follow R. In contrast, Holladay et al. [9] address both concerns by con-



(a) (b) (c)

Fig. 1: (a) CTR with coin for scale. (b) Depiction of a CTR deployed via the sinus. A reference path R, used to cut a window in the skull with the CTR's tip, is depicted in yellow. (c) An example of a surgical system that uses our planner as a core component.



(a) (b)

Fig. 2: (a) Layered graph L and (b) nearest-neighbor graph N_k constructed in $\mathcal{C}$-space to follow a reference path R in task-space. Note that a layer in L is depicted as a vertical stack of nodes. If reaching r_i is impossible, all paths in L become blocked. The structure of N_k, however, can bypass such blockages.

structing a layered graph L in $\mathcal{C}$-space and searching it for a path that *directly* minimizes the Fréchet distance [21] from R.

Key to this algorithm's efficiency are the graph's ordered layers (Fig. 2a), each consisting of multiple Inverse Kinematics (IK) solutions for a specific point on R. Directed edges connect each configuration to others in the same and adjacent layers, ensuring monotonic progress. Unfortunately, this requires any path through L to touch R at *every* layer, leading to failure in constrained environments such as ours. For example, a layer mapping to a point $r_i \in R$ made unreachable due to collisions causes no collision-free path to exist in the graph. The CTR's kinematics make it difficult for a surgeon specifying a path to perfectly distinguish feasible locations in task space, accentuating this failure mode.

Our key insight is to generalize this algorithm by searching over a *nearest-neighbor* graph, with less restrictive connections that allow deviation from infeasible portions of R (Fig. 2b). This makes planning in constrained anatomical environments possible, and thus enables our key contribution—a system implemented on a CTR, presented with a set of simulated and physical experiments. The former reveal that our planner produces higher-quality solutions more reliably than alternative algorithms, while the latter inform algorithmic next-steps necessary to bridge the simulation and real-robot gap.

2 Technical Approach

2.1 Notation and Definitions

The Fréchet distance is a metric of path similarity that has been extensively studied in computational geometry, with varied applications such as speech [12] and handwriting [18] recognition. It can be explained intuitively via an analogy where a dog on a leash traverses one path while its owner traverses another, each with independent speed parameterizations α and β. In this case, the Fréchet distance is the shortest leash length required for the two to stay connected, assuming the dog and its owner are selecting optimal values of α and β.

Motion plans for surgical tasks, such as tissue manipulation, must respect the physician's intent by (i) minimizing the tip's deviation from the specified

path at each point in time and (ii) following the "flow" of the path by ensuring that consecutive points along the two paths are traversed in the same order. Because the Fréchet distance captures both objectives, it provides an excellent optimization criteria for our domain.

Our planner searches for these motion plans in $\mathcal{C}$-space, the set of all possible configurations of the robot. Each configuration is a d-dimensional point that uniquely defines the robot's shape. Similarly, the task-space of the robot is defined as the space of all possible end-effector positions. We use the Forward Kinematics (FK) operator to map points in $\mathcal{C}$-space to points in task-space, and use the Inverse Kinematics (IK) operator to perform the reverse mapping.

For a CTR, a configuration must describe the translation ℓ_i and rotation θ_i of each individual tube. Thus, for a CTR comprised of k tubes (typically three or four), each configuration is a $2k$-length tuple $q = (\ell_1, \theta_1, \ldots, \ell_k, \theta_k)$. We define the robot's task-space as the $\mathbb{R}^3$ location of its tip, which is sufficient for many procedures such as cutting with heat or a laser. However, we note that our planner can easily be used with other definitions of task-space that also account for the tips's orientation.

In our setting, the physician specifies a reference path R in task-space as a sequence of $\mathbb{R}^3$ waypoints. Let Γ denote the set of collision-free $\mathcal{C}$-space paths and $\mathcal{F} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ denote the discrete[4] Fréchet distance between two task-space paths. Our problem now calls for computing

$$\arg\min_{\gamma \in \Gamma} \mathcal{F}(\mathrm{FK}(\gamma), R).^{5}$$

2.2 Algorithmic Approach

Recall that Holladay et al. [9] construct a layered graph L in $\mathcal{C}$-space. This is done by evenly sampling waypoints along the reference path R, and computing a "layer" of distinct IK solutions for each one. Each of these configurations has out-edges to all configurations in both the same and next immediate layer. Following Har-Peled and Raichel [8], the cross-product graph $\Phi = L \times R$[6] is then constructed and searched for a minimal-bottleneck path, which induces the path $\xi \in L$ such that $\mathrm{FK}(\xi)$ minimizes the discrete Fréchet distance with R.

Algorithmic Enhancement: Our early experiments, however, revealed that this planner often fails in surgical scenarios. In these constrained environments, some sampled waypoint $r_i \in R$ may be unreachable, often due to a combination of collisions and robot kinematics (Fig. 3). Even the existence

[4] Computing the continuous Fréchet distance, is notoriously difficult [16]. Thus, we use the *discrete* variant, easily computed using dynamic programming [5]. Here, the "leash" between points on the two paths is computed only for a discrete set of points and serves as an approximation of the (continous) Fréchet distance.

[5] By a slight abuse of notation we use FK to map both points as well as paths in $\mathcal{C}$-space to points and paths in task space, respectively.

[6] By a slight abuse of notation we treat R both as the discrete reference path as well as the one-dimensional graph defined by this sequence of waypoints.

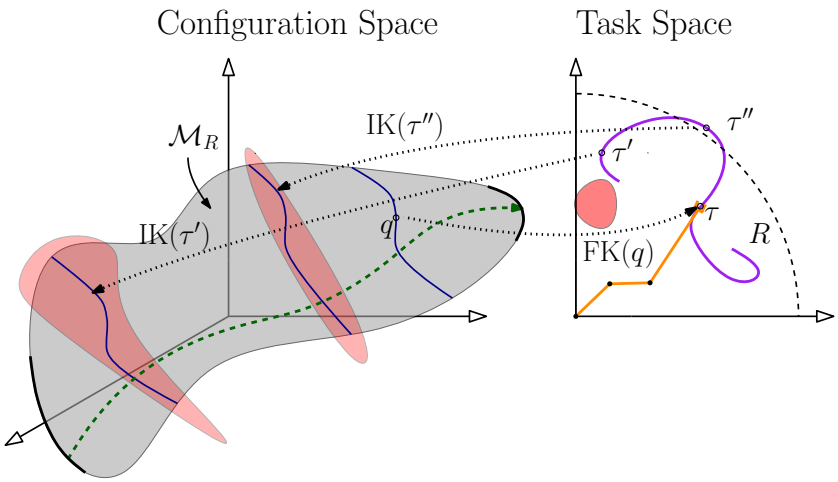


Fig. 3: A reference path R (purple, right) in the task-space of a robot (yellow, right) induces a manifold $\mathcal{M}_R$ of IK solutions in $\mathcal{C}$-space (grey, left). Following R via a layered graph requires traversing $\mathcal{M}_R$ from one end to the other, such as by taking the path in green. However, even if R is not directly in collision with some obstacle (red, right), its presence may cause a point $\tau' \in R$ to map to a self-manifold [3] IK(τ') that is completely in collision (red, left) given the kinematics of the robot. These in-collision self-manifolds require deviating from $\mathcal{M}_R$ to approximate R, and make traversal via the layered graph impossible.

of *one* such waypoint prevents the algorithm from finding any solutions, as all paths are blocked by a layer of configurations in collision (Fig. 2a). To prevent this, we construct a nearest-neighbor graph N_k by uniformly sampling IK solutions from R and connecting each to its k-nearest neighbors (NN) in $\mathcal{C}$-space. This less-rigid structure allows the planner to avoid IK solutions sampled from infeasible waypoints (Fig. 2b).

Specifically, each configuration q is connected to its k-NN in $Q_s(q)$, the set containing all $q_s \in N_k$ such that FK(q_s) lies after FK(q) along R. This ensures that any path through N_k monotonically follows R. As in [9], we generate the path $\xi \in N_k$ that minimizes the Fréchet distance with R by searching the cross-product graph $\Phi = N_k \times R$.

Densification: This graph-based approach allows solutions to be improved on-demand by sampling additional IK solutions from R and densifying N_k. To connect each newly-sampled solution q to N_k, we add out-edges from q to its k-NN in $Q_s(q)$. In addition, we add in-edges to q from its k-NN in $Q_p(q)$, the set containing all $q_p \in N_k$ such that FK(q_p) lies before FK(q) along R. These additional connections increase the set of paths the planner can search over, while still constraining it to monotonically follow R.

Implementation details: We collision check edges lazily [4,7], using LPA* [11] for efficient replanning. Our CTR has three tubes, each of which can be rotated and translated, inducing a 6-dimensional $\mathcal{C}$-space. We use the damped

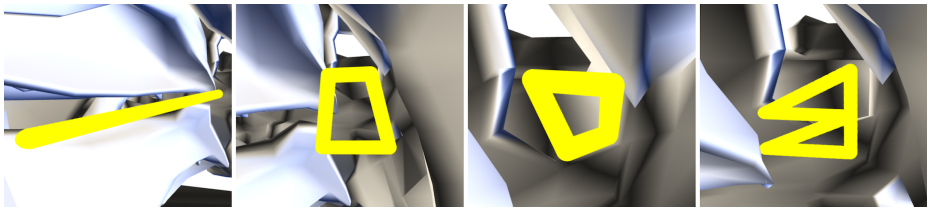


Fig. 4: From left to right, reference paths (i) through (iv) used in our experiments.

least squares (DLS) IK algorithm [20], which has previously been used with the CTR [19]. We represent obstacles as polygonal meshes, and use FCL [13] for collision checks. We define the distance between two configurations q_1 and q_2 as:

$$\sum_k |\ell_{k1} - \ell_{k2}| + \rho(\theta_{k1}, \theta_{k2})$$

where ρ denotes the $SO(2)$ difference between two angles. We note that the choice of distance metric can drastically affect the algorithm's performance [1].

3 Experiments and Results

We consider four surgically-motivated reference paths in our experimental evaluation (Fig. 4), all within a human skull base. The first (i) moves the CTR's tip in a straight line towards the back of the sinuses, ending near a region where the second (ii) traces a trapezoid. These two paths are inspired by a clinical scenario where the CTR is deployed via the first path, and then uses the second to cut a window allowing deeper access into the skull. We also test (iii) a different version of the second path, as well as (iv) a W-shaped path that could be used for wound irrigation. While not in collision with the skull, these last two paths reside in constrained areas and thus exhibit the phenomenon depicted in Fig. 3.

3.1 Simulation Comparison

We evaluate our algorithm, which we term NN Graph, in simulation and compare its performance with the following alternatives: (a) VFP which follows a vector-field along R by using the Jacobian to integrate a path in $\mathcal{C}$-space [17], (b) GreedyIK which adaptively samples a set of ordered IK solutions from R and attempts to interpolate a collision-free path in $\mathcal{C}$-space between them [17], (c) CBiRRT which enables planning on constraint manifolds [2] and (d) Layered Graph—the approach presented by Holladay et al. [9]. It is worth noting that planners (a) and (b) are myopic but efficient and that all but (d) were not designed to minimize the Fréchet distance from a reference path.

Simple Paths: We first compare the planners using paths (i) and (ii), both without regions made unreachable by the collision geometry or CTR kinematics. We report the Fréchet distances of the produced trajectories from R for a wide set of parameters (Fig. 5) and average results over ten random seeds.

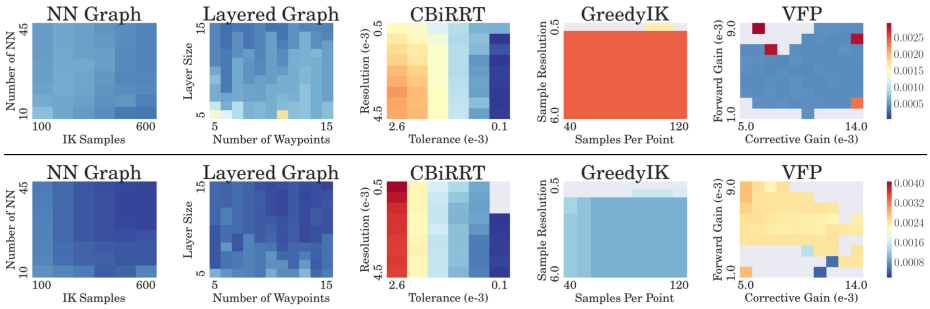


Fig. 5: Fréchet distances over a range of parameters for path (i) at top and path (ii) at bottom. A grey square indicates the algorithm failed to find a solution with that combination of parameters.

We see that for both paths, all planners successfully produce solutions. When using path (i), a straight line, performance for all planners remains fairly stable as parameters vary. However, both paths reveal weaknesses in VFP and GreedyIK. With the former, optimal performance on path (ii) is only achieved under two sets of parameters. Similarly, GreedyIK produces results of markedly lower quality than NN Graph across most parameters. CBiRRT, meanwhile, is used to pin the CTR's tip within a tolerance of R and succeeds when this tolerance is made sufficiently tight. This is especially evident on path (i), the straight line.

Constrained Paths: We also compare the planners using paths (iii) and (iv), as seen in Fig. 6. Due to being located in constrained areas, these paths contain small regions infeasible for the CTR to follow perfectly. We note that these paths were not hand-engineered adversarially, but were generated by a human familiar with the CTR and real-world surgical procedures.

Due to the infeasible regions, Layered Graph was unable to find any solutions with either path, confirming the failure mode we observed in our early experiments. While CBiRRT still constrains the CTR's tip to lie within a tolerance of R, it lacks any notion of path following, causing it to violate the flow of these more constrained paths. This oblivious nature also hinders GreedyIK, causing it to display unstable behavior where sampling *more* IK solutions can cause a sharp drop in solution quality.

In our experiments with all four paths, NN Graph displayed stable behavior under parameter variation and was the only planner able to consistently produce high-quality solutions. In fact, for each of the four other planners, at least one path led to either very low quality solutions or no solutions being found.

Additional Comparisons: We use our results from path (iii) to perform additional comparisons between planners. In the first of these, we correlate Fréchet distance and runtime for each planner by averaging results across all parameters over ten-second windows (Fig. 7a). This confirms that GreedyIK produces solutions quickly due to its simple nature. However, our planner outperforms the others, which display fluctuating behavior over time.

In addition to evaluating our planner over a range of parameters, we are interested in how a *single* set of parameters generalizes across a range of reference

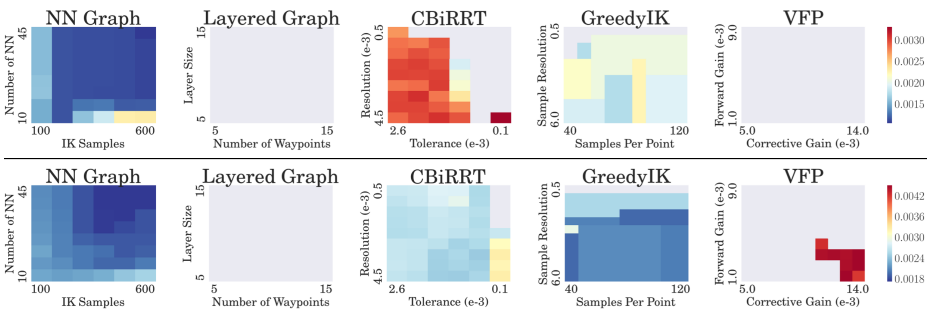


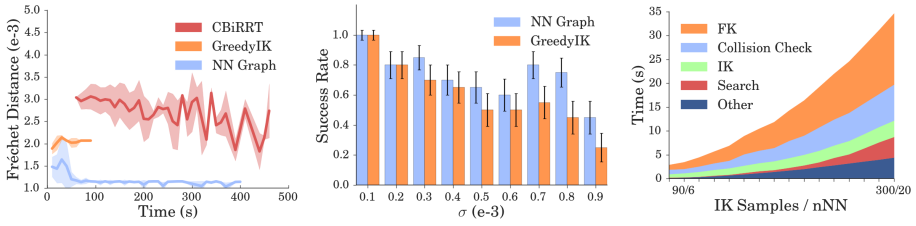
Fig. 6: Fréchet distances for path (iii) at top and path (iv) at bottom. We note that Layered Graph failed on both paths, and that VFP failed on path (iii).

paths. We compare against GreedyIK using its optimal parameters, and select a set for NN Graph that lead to similar runtime on path (iii). Generalization is evaluated by adding independent Gaussian noise to the corners defining this path and recording planner success rates (each taken over 20 trials) under increasing levels of noise (Fig. 7b). We see that as the level of noise increases, our planner has a higher success rate than GreedyIK.

3.2 Algorithmic Behavior

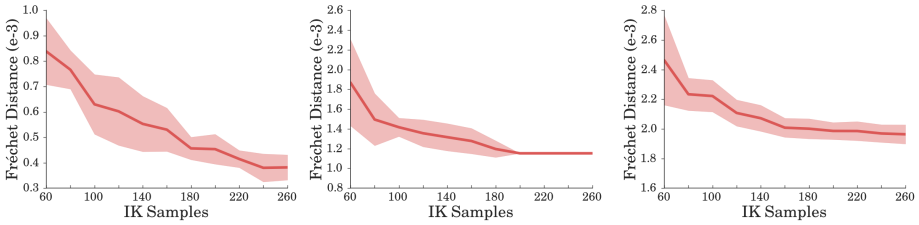
Runtime Analysis: A decomposition of NN Graph's runtime on path (iii) as we linearly scale both the number of nearest-neighbors and IK solutions sampled is shown in Fig. 7c. FK, used in constructing the nodes of the cross-product graph Φ [8,9], is notably the most expensive component. This cost arises from modeling complex interactions between tubes, which requires solving a series of differential equations rather than the typical matrix multiplication.

Densification: We also explore the ability of densification to improve existing NN Graph solutions. This provides our planner an advantage over alternatives, as it enables a physician to improve a solution online until they are satisfied. Our trials use 20 nearest-neighbors and augment N_k with an additional 20 configurations per densification iteration. We report the Fréchet distances of solutions as the graph increases in size.



(a) Fréchet vs. Time (b) Robustness to Noise (c) Runtime Components

Fig. 7: (a) Fréchet distance as a function of the runtime for each planner. (b) Success rates for NN Graph and GreedyIK as we inject more noise into R. (c) A decomposition of the runtime of our NN Graph. All results use path (iii)



(a) Path (ii) (b) Path (iii) (c) Path (iv)

Fig. 8: Densification results.

Densification provides very little improvement with path (i), likely because this simple straight-line path makes the initial solution already close to optimal. The other three paths display notable improvement as the size of N_k increases (Fig. 8). However, paths (iii) and (iv) both exhibit diminishing returns past a given iteration.

3.3 Physical Experiments

We evaluate our algorithm on the real robot, testing all four paths inside a 3D-printed model of the skull base anatomy. Parameters are selected that create high-quality solutions in simulation. To measure Fréchet distance, we use a magnetic tracking system to localize the tip during execution. Because this system is noisy, we execute each trajectory ten times and average the results.

Imperfections in our kinematic model (discussed in the next section) prevent us from scoring path (iii) due to collisions. While this path was not executed successfully, the other three were. Paths (iv), (ii), and (i) were each executed with average real Fréchet errors of 0.00806, 0.00747, and 0.00744 meters respectively. These are all significantly higher than the respective simulation errors of 0.00211, 0.00031, and 0.00034 meters. This discrepancy arises primarily from the inaccuracies in our kinematic model. A comparison of simulated and real execution using path (ii) is shown in Fig. 9.

4 Experimental Insights and Future Work

Our experiments support our use of a nearest-neighbor graph and demonstrate the advantages of our planner over alternatives. They also reveal previously unconsidered domain-specific challenges, which we can only now begin to address. We propose solutions to each of these, which we will explore in future work.

Our planner's high runtime is mainly due to our robot's expensive FK (Fig. 7c), motivating a paradigm shift in planner design for all robots that incur high cost in this operation. To address this, we propose constructing the cross-product graph Φ implicitly, which avoids unnecessarily invoking FK to create nodes never expanded during the search.

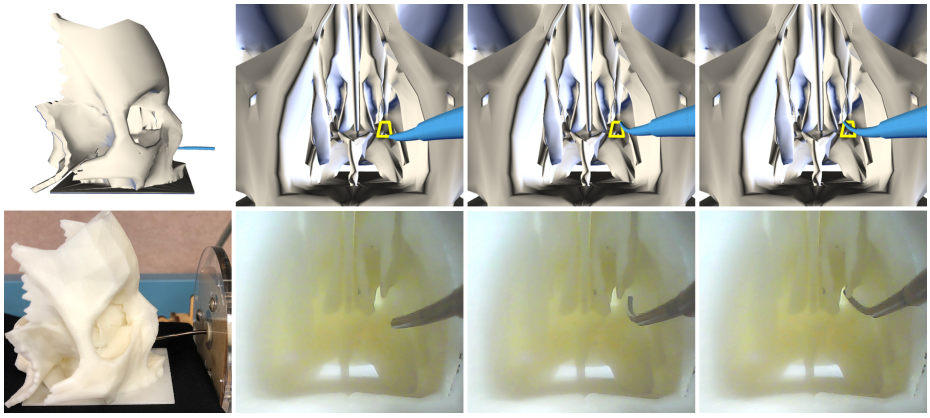


Fig. 9: Simulated (top) and real (bottom) execution of path (ii). The reference path is colored yellow in simulation.

While we demonstrate a successful planner, our physical experiments reveal the kinematic model has now become the limiting factor. Although this model is state-of-the-art [15], it fails to perfectly model certain physical phenomenon, such as friction between tubes. This causes the shape of the robot in simulation to only approximate reality. Naturally, improving this model is a solution. We also propose taking its uncertainty into account via a bicriteria optimization problem of (i) minimizing Fréchet and (ii) maximizing the distance of the robot from obstacles to prevent collisions. This has the added benefit of mitigating error in the collision geometry, constructed from a preoperative volumetric scan [10].

Additionally, successful execution of path (ii) on the real CTR required manually tuning the insertion point of the robot into the skull. The kinematics of the robot induce a trumpet-shaped workspace, leading to greater range of motion farther from this insertion point. One of the insertion points was too close to the reference path, placing it outside the robot's workspace and causing a planning failure. Likewise, the successful insertion point placed the shaft of the robot in a less-constrained area, preventing a collision that otherwise occurred due to modeling error. Because it proved so consequential, we propose incorporating this selection of insertion point as a planning subproblem.

Acknowledgments: We thank Bob Webster and his group at Vanderbilt University for numerous discussions on CTRs and for creating the CTR used here. We thank Rachel Holladay for her invaluable insight in discussing her previous work. This work was (partially) funded by the National Institute of Health R01 (#R01EB019335), National Science Foundation CPS (#1544797), National Science Foundation NRI (#1637748), National Science Foundation RI Award 1149965, the Office of Naval Research, the RCTA, Amazon, and Honda.

References

1. A. Atias, K. Solovey, O. Salzman, and D. Halperin. Effective metrics for multi-robot motion-planning. *CoRR*, abs/1705.10300, 2017.

2. D. Berenson, S. Srinivasa, D. Ferguson, and J. Kuffner. Manipulation planning on constraint manifolds. In *ICRA*, pages 625–632, 2009.
3. J. Burdick. On the inverse kinematics of redundant manipulators: characterization of the self-motion manifolds. In *ICRA*, pages 264–270, 1989.
4. C. Dellin and S. Srinivasa. A unifying formalism for shortest path problems with expensive edge evaluations via lazy best-first search over paths with edge selectors. In *ICAPS*, pages 459–467, 2016.
5. T. Eiter and H. Mannila. Computing discrete fréchet distance. Technical report, Citeseer, 1994.
6. H. Gilbert, D. Rucker, and R. J. Webster III. Concentric tube robots: The state of the art and future directions. In *ISRR*, pages 253–269, 2013.
7. N. Haghtalab, S. Mackenzie, A. Procaccia, O. Salzman, and S. Srinivasa. The provable virtue of laziness in motion planning. In *ICAPS*, pages 106–113, 2018.
8. S. Har-Peled and B. Raichel. The Frećhet distance revisited and extended. *TALG*, 10(1):3, 2014.
9. R. Holladay, O. Salzman, and S. Srinivasa. Minimizing task space Frećhet error via efficient incremental graph search. *CoRR*, abs/1710.06738, 2017.
10. H. Johnson, M. McCormick, and L. Ibanez. *The ITK Software Guide Book 1: Introduction and Development Guidelines - Volume 1*. Kitware, Inc., USA, 2015.
11. S. Koenig, M. Likhachev, and D. Furcy. Lifelong planning A*. *Artif. Intell.*, 155(1-2):93–146, 2004.
12. S. Kwong, Q.H. He, K.F. Man, K.S. Tang, and C.W. Chau. Parallel genetic-based hybrid pattern matching algorithm for isolated word recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 12(05):573–594, 1998.
13. J. Pan, S. Chitta, and D. Manocha. FCL: A general purpose library for collision and proximity queries. In *ICRA*, pages 3859–3866, 2012.
14. ROS Industrial. Descartes, 2015. Last updated: 20 April 2018.
15. D. Rucker. *The mechanics of continuum robots: model-based sensing and control*. PhD thesis, Vanderbilt University, 2011.
16. K. Solovey and D. Halperin. Sampling-based bottleneck pathfinding with applications to fréchet matching. In *European Symposium on Algorithms, ESA*, pages 76:1–76:16, 2016.
17. S. Srinivasa, A. Johnson, G. Lee, M. Koval, S. Choudhury, J. King, C. Dellin, M. Harding, D. Butterworth, P. Velagapudi, and A. Thackston. A system for multi-step mobile manipulation: Architecture, algorithms, and experiments. In *ISER*, pages 254–265, 2016.
18. E. Sriraghavendra, K. Karthik, and C. Bhattacharyya. Fréchet distance based approach for searching online handwritten documents. In *Document Analysis and Recognition*, volume 1, pages 461–465. IEEE, 2007.
19. L. Torres, A. Kuntz, H. Gilbert, P. Swaney, R. Hendrick, R. J. Webster III, and R. Alterovitz. A motion planning approach to automatic obstacle avoidance during concentric tube robot teleoperation. In *ICRA*, pages 2361–2367, 2015.
20. C. Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man and Cybernetics*, 16:93 – 101, 02 1986.
21. T. Wylie et al. *The discrete Fréchet distance with applications*. PhD thesis, Montana State University-Bozeman, College of Engineering, 2013.
22. Z. Yao and K. Gupta. Path planning with general end-effector constraints: Using task space to guide configuration space search. In *IROS*, pages 1875–1880, 2005.