

Planning Fireworks Trajectories for Steerable Medical Needles to Reduce Patient Trauma

Jijie Xu, Vincent Duindam, Ron Alterovitz, Jean Pouliot, J. Adam M. Cunha, I-Chow Hsu and Ken Goldberg

Abstract—Accurate insertion of needles to targets in 3D anatomy is required for numerous medical procedures. To reduce patient trauma, a “fireworks” needle insertion approach can be used in which multiple needles are inserted from a single small region on the patient’s skin to multiple targets in the tissue. In this paper, we explore motion planning for “fireworks” needle insertion in 3D environments by developing an algorithm based on Rapidly-exploring Random Trees (RRTs). Given a set of targets, we propose an algorithm to quickly explore the configuration space by building a forest of RRTs and to find feasible plans for multiple steerable needles from a single entry region. We present two path selection algorithms with different optimality considerations to optimize the final plan among all feasible outputs. Finally, we demonstrate the performance of the proposed algorithm with a simulation based on a prostate cancer treatment environment.

I. INTRODUCTION

Accurately inserting a needle to a specific target in a patient’s anatomy is difficult due to lack of maneuverability, limited visibility, and possible obstructions between the needle entry point and the target. A new class of steerable needles, composed of a highly flexible material and with a bevel-tip, are capable of following controlled, curved trajectories, enabling them to reach previously inaccessible targets while avoiding anatomical obstacles [17], [2], [1]. Because the needle bends in the direction of the bevel, the feasible paths of these needles can be approximated by concatenating constant-curvature arcs. By enabling avoidance of sensitive structures, steerable needles have the potential to reduce patient trauma that can result from the needle puncturing or cutting sensitive, healthy tissues.

In this paper, we explore a new approach to further reduce patient trauma for cases in which the physician must reach multiple targets during the same procedure. We use a “fireworks” trajectory [6] in which needle paths are computed to multiple targets from a single entry region or port in the

This work was supported by NIH grant R01 EB006435 and the Netherlands Organization for Scientific Research.

J. Xu is with the B. Thomas Golisano College of Computing and Information Science, Rochester Institute of Technology, USA. (e-mail: jijie.xu@rit.edu)

V. Duindam was with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA.

R. Alterovitz is with the Department of Computer Science, University of North Carolina at Chapel Hill, USA. (e-mail: ron@cs.unc.edu)

J. A. M. Cunha, I.-C. Hsu, and J. Pouliot are with the Helen Diller Family Comprehensive Cancer Center, University of California, San Francisco, USA.

K. Goldberg is with the Department of Industrial Engineering and Operations Research and the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA. (e-mail: goldberg@berkeley.edu)

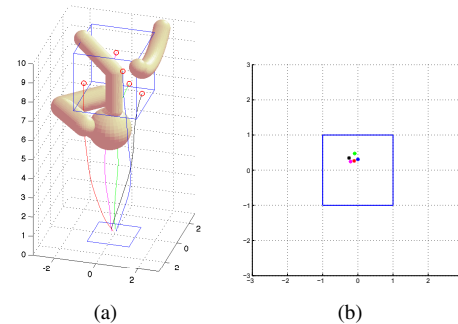


Fig. 1. An example of “fireworks” steerable needle insertion: 5 needles are inserted from very small entry region to reach 5 targets in the workspace while avoiding collision with the obstacles.

patient’s skin. Planning motions for steerable needles in this context requires that the needle paths reach multiple targets from the same entry region while avoiding obstacles and satisfying the kinematic constraints of the needles’ motion.

We explore motion planning of “fireworks” insertion of steerable needles by extending a previously developed sampling-based motion planning method [18]. The steerable needle planner considers 3D environments with obstacles, enabling modeling of patient-specific anatomical obstacles extracted from medical images. The approach is inspired by Rapidly-exploring Random Trees (RRTs); our algorithm quickly grows a forest of RRTs to explore the configuration space and find feasible paths for all targets. Computing paths to all targets simultaneously can accelerate motion planning compared to executing a separate planner for each target, e.g. by reusing samples and reducing collision detections computed. Our approach also leverages the fact that the entry location is defined as a region rather than a point; due to the kinematic constraints of steerable needles, it may not be possible to reach all targets from a single entry point. Our method considers a surface corresponding to an entry port on the patient’s skin, and minimizes the size of this surface while still reaching all targets.

We present two path selection algorithms to optimize the final plan among feasible outputs. We evaluate the algorithm using a 3D environment inspired by prostate cancer brachytherapy treatment, a type of radiation treatment in which physicians use needles to implant radioactive sources inside the prostate.

II. RELATED WORK

The use of a bevel tip for a medical needle significantly affects the bending forces acting on the needle during insertion [13]. Based on this observation, Webster et al. [16] developed and experimentally verified a nonholonomic model of the needle's motion in stiff tissues based on a generalized bicycle model. This model can be approximated by arcs of constant curvature in the direction of the bevel-tip. The radius of that curvature is not significantly affected by the insertion velocity [17].

There has been much work on motion planning for steerable bevel-tip needles in a 2D workspace that incorporates the effects of tissue deformations and motion uncertainties. Alterovitz et al. [2] addressed the challenge of accurately steering a needle around obstacles to a target in deformable tissue. They formulate the motion planning problem as a nonlinear optimization problem that uses a simulation of tissue deformation during needle insertion as a function in an optimization that minimizes needle placement error. To consider motion uncertainties due to needle/tissue interaction, Alterovitz et al. efficiently discretized the state space of the needle and formulated the motion planning problem as a Markov Decision Process (MDP) to maximize the probability of successfully reaching the target [3], [1]. Alterovitz et al. also generalized this approach into the Stochastic Motion Roadmap (SMR), a sampling-based framework for planning under motion uncertainty [4].

Recent methods have considered more complex 3D environments. Park et al. developed a diffusion-based motion planning algorithm to numerically compute a path [14]. By representing the bevel-tip needle's 3D motion as a screw motion, Duindam et al. [7] formulated the 3D motion planning problem for steerable needles as an optimization problem with a discretized control space and incorporated obstacle avoidance. Duindam et al. also developed a fast motion planner using a geometric approach that is not guaranteed to find a solution when one exists [8]. Inspired by the Rapidly-exploring Random Tree (RRTs) algorithms, Xu et al. [18] developed a 3D sampling-based motion planner that efficiently builds a global tree to quickly and probabilistically explore the entire workspace and search for a feasible plan.

The method of Rapidly-exploring Random Trees (RRTs), which was first introduced by LaValle [10], is a successful roadmap-based motion planning method that has shown its potential in dealing with motion planning problems for nonholonomic systems [11]. It incrementally grows a tree toward the target configuration by searching feasible paths in the configuration space, and provides an efficient and quick search in complex environments of high dimensions with different constraints [11]. Based on the original RRTs structure, many variants have been developed to improve the efficiency of searching [9], extend RRTs to more complex configuration space (\mathcal{C} -space) [5], and enhance the ability to explore difficult regions in searching the environment [15]. In this paper, we extend our RRT-inspired algorithms developed in [18] and propose an RRT-forest exploration strategy

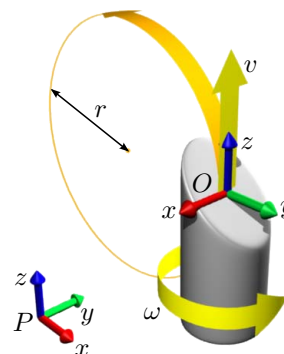


Fig. 2. Model of the bevel-tip needle.

together with a plan selection algorithm to efficiently solve the motion planning problem for the “fireworks” insertion.

III. PROBLEM STATEMENT

The problem we address is to compute motion plans for multiple steerable needles from a single entry region to multiple targets. We assume that all needles satisfy the properties of bevel-tip steerable needles: each needle is sufficiently flexible such that rotating the needle at the base will not change its position in the workspace and each needle's motion through the tissue is fully determined by its tip, meaning that the needle shaft follows the path cut by the needle tip. We also assume that all needles used in a procedure have identical properties (i.e. follow paths with the same curvatures). In our current implementation, we define the feasible workspace is a stiff 3D cuboid and all spatial obstacles are approximated using 3D spheres with various radii to simplify computational expense for collision detection. (Multiple needle insertion planning in a deformable environment and with more geometrically complex obstacles will be considered in future work.)

Given an entry region and a set of target configurations, the problem is to determine a set of feasible paths and the corresponding sequences of controls (insertion depths and rotations for each steerable needle) so that each target can be reached by a needle tip inserted from the entry region while avoiding obstacles and staying inside the workspace. Formally, the inputs and outputs are:

Input: Boundaries of the workspace, parameters of all steerable needles, obstacles locations, an entry region in the entry surface to insert all needles, and a set of target coordinates to reach.

Output: For each target coordinate, a needle path defined by its entry point and a sequence of controls, or a report that no path was found.

IV. KINEMATICS OF BEVEL-TIP FLEXIBLE NEEDLES

We summarize the kinematics of bevel-tip steerable needles; for more details see [17], [7]. Consider the bevel-tip needle shown in Fig. 2. Referring to the notation in [12], attach a spatial frame P to the base of the needle and a body frame O to the geometric center of the needle's bevel tip, respectively. The configuration of the needle tip can be

represented by the 4-by-4 transformation matrix of the object frame relative to the spatial frame,

$$g_{PO} = \begin{bmatrix} R_{PO} & p_{PO} \\ 0 & 1 \end{bmatrix} \in SE(3),$$

where $R_{PO} \in SO(3)$ is the rotation matrix and $p_{PO} \in T(3)$ is the position of frame O relative to frame P .

The motion of the needle is fully determined by two motions of the bevel tip: insertion with velocity $v(t)$ in the z direction and rotation with velocity $\omega(t)$ about the z axis of the body frame O [16], [7]. It has been experimentally shown [17] that the bevel-tip needle will approximately follow a curved path with constant curvature $\kappa = \frac{1}{r}$ while pushed with zero bevel rotation velocity, i.e. $\omega = 0$. The instantaneous velocity of the needle tip can be represented in the body frame O as

$$V_{PO}^b = [\mathbf{v}^T \quad \mathbf{w}^T]^T = [0 \quad 0 \quad v(t) \quad v(t)/r \quad 0 \quad \omega(t)]^T. \quad (1)$$

When V_{PO}^b is constant, i.e., $v(t)$ and $\omega(t)$ are constant, the configuration of the needle tip relative to the spatial frame after being pushed for a time interval t is

$$g_{PO}(t) = g_{PO}(0)e^{\hat{V}_{PO}^b t}, \quad (2)$$

where $g_{PO}(0)$ is the initial configuration of the needle frame relative to the spatial frame, and

$$\hat{V}_{PO}^b = \begin{bmatrix} 0 & -\omega(t) & 0 & 0 \\ \omega(t) & 0 & -v(t)/r & 0 \\ 0 & v(t)/r & 0 & v(t) \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3)$$

According to the kinematics analysis in [7], the trajectory of the needle only depends on the ratio ω/v but not on the values of the two individual terms. Therefore, we discretize the entire insertion into N steps with N time segments $\{I_1, \dots, I_N\}$ and assume that the velocity in each single step, $V_{PO}^b(I_n)$, is constant. The final needle tip configuration can be computed as a product of exponentials [12], [7]

$$g_{PO}(T) = g_{PO}(0)e^{\hat{V}_{PO}^b(I_1)I_1} \dots e^{\hat{V}_{PO}^b(I_N)I_N}. \quad (4)$$

Note that the bevel-tip needle can only follow curved paths with a curvature $\kappa = 1/r$, as shown in Fig. 2. For this reason, reachable configurations of the needle are locally constrained inside the volume of a region (see Fig. 3) defined by

$$p_z \geq \sqrt{2r \sqrt{p_x^2 + p_y^2} - p_x^2 - p_y^2}, \quad (5)$$

where (p_x, p_y, p_z) are coordinates of a point in the body frame O .

V. MOTION PLANNING FOR MULTIPLE STEERABLE NEEDLES

A. “Fireworks” needle insertion planning

The configuration of the needle tip is determined by its position (x, y, z) together with its three Euler angles (ϕ, θ, ψ) . However, the insertion task only requires the needle to reach a specific position in the 3 dimensional workspace, equivalent to \mathbb{R}^3 . Following its kinematics (2), the needle’s

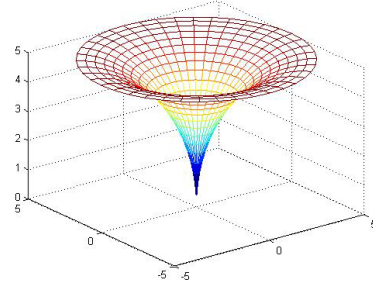


Fig. 3. The reachable region of local needle motion.

```

BUILD_FOREST( $S_{init}, S_{goal}$ )
1. for all  $s_{goal}^i \in S_{goal}$ 
2.  $\mathcal{T}_i = \mathcal{T}_{init}(s_{goal}^i)$ 
3.  $\mathcal{F}_{add\_tree}(\mathcal{T}_i)$ 
4. while  $No\_of\_Iteration < Max\_Iteration$ 
5.    $s_{rand} \leftarrow RANDOM\_STATE()$ 
6.    $\mathcal{F} \leftarrow EXTEND\_FOREST(\mathcal{F}, s_{rand})$ 
7.  $p^* = SELECT\_PATHS(\mathcal{F})$ 
8. RETURN  $p^*$ 

```

TABLE I

THE SCENARIO OF ALGORITHM 1.

trajectory to reach a target at $g_{PO}(t)$ can be computed backwardly from its target as

$$g_{PO}(t - \delta t) = g_{PO}(t)e^{-\hat{V}_{PO}^b \delta t}. \quad (6)$$

Given the workspace’s boundaries, $([x_{min}, x_{max}], [y_{min}, y_{max}], [z_{min}, z_{max}])$, the obstacles information, the target locations s_{goal} and the specified entry zone S_{entry} , a motion planning algorithm based on RRTs with backchaining has been developed to quickly explore the configuration space from the target to find feasible solutions for insertion tasks with single needle and single target (SNST) [18].

For “fireworks” needle insertions, one can separately execute the algorithm for SNST for each individual target. However, repeated sampling procedures for each needle unnecessarily increases the computational cost. Moreover, separated collision detection for a needle in an environment with previous solutions of other needles also requires extensive computational cost. By extending that algorithm, we propose a new algorithm to quickly grow a forest of RRTs to search feasible paths for all targets simultaneously.

Algorithm 1 (*Forest of RRTs with backchaining*): For all $s_{goal}^i \in S_{goal}$, initialize the forest \mathcal{F} by initializing all trees \mathcal{T}_i rooting at s_{goal}^i . Randomly sample a state s_{rand} in the collision free configuration space \mathcal{CS}_{free} . For all trees in the forest, the reachable neighbor test is executed for all nodes $s_i \in \mathcal{T}_i$ to find a reachable set S_{reach}^i , which contains all nodes reachable from s_{rand} . The nearest neighbor search is executed inside S_{reach}^i to find the nearest neighbor of s_{rand} , denoted by $s_{near}^i \in S_{reach}^i$, which has the shortest distance to s_{rand} . The distance used in the nearest neighbor search can be defined in different ways by defining different metrics

```

RANDOM_STATE()
1.  $p = rand(0, 1)$ 
2. if  $p < p_1$ 
3.    $s_{rand} = \text{UNIFORM\_SAMPLE}(S_{init})$ 
4. if  $p_1 < p < p_2$ 
5.    $s_{rand} = \text{UNIFORM\_SAMPLE}(S_{goal})$ 
6. else
7.    $s_{rand} = \text{UNIFORM\_SAMPLE}(CS_{free}/(S_{init} \cup S_{goal}))$ 
8. RETURN  $s_{rand}$ 

```

```

EXTEND_FOREST( $\mathcal{F}, s_{rand}$ )
1. for all  $\mathcal{T}_i \in \mathcal{F}$ 
2.   EXTEND_TREE( $\mathcal{T}_i, s_{rand}$ )
3. RETURN  $\mathcal{F}$ 

```

```

EXTEND_TREE( $\mathcal{T}, s_{rand}$ )
1.  $s_{reach}^i \leftarrow \text{REACHABLE\_NEIGHBORS}(\mathcal{T}, s_{rand})$ 
2.  $s_{near} \leftarrow \text{NEAREST\_NEIGHBOR}(s_{reach}^i, s_{rand})$ 
3.  $(s_{new}, u_{new}) \leftarrow \text{NEW\_STATE}(s_{near}, s_{rand}, \mathcal{U})$ 
4.  $\mathcal{T}.\text{add\_vertex}(s_{new})$ 
5.  $\mathcal{T}.\text{add\_edge}(s_{near}, s_{new}, u_{new})$ 
6. RETURN  $\mathcal{T}$ 

```

```

REACHABLE_NEIGHBORS( $\mathcal{T}, s_{rand}$ )
1. For all  $s_i \in \mathcal{T}$ 
2. if  $s_i$  is reachable from  $s_{rand}$ 
3.   add  $s_i$  to  $S_{reach}$ 
4. RETURN  $S_{reach}$ 

```

```

NEW_STATE( $s_{near}, s_{rand}, \mathcal{U}$ )
1.  $\mathcal{U}_{rand} \leftarrow \text{CONTROL\_SAMPLING}(\mathcal{U})$ 
2. FOR all  $u_i \in \mathcal{U}_{rand}$ 
3.    $s_{new}^i(i) = s_{near} + F_{d_{near}}(s, -u_i)\delta t$ 
4.  $S_{new} = \cup_i s_{new}^i(i)$ 
5.  $s_{new} \leftarrow \text{NEAREST\_NEIGHBOR}(S_{new}, s_{rand})$ 
6.  $u_{new} = -u_i$  such that  $s_i = s_{new}$ 
7. RETURN  $s_{new}, u_{new}$ 

```

TABLE II
DETAILED PROCEDURE OF ALGORITHM 1.

on the configuration space. Then we uniformly sample the negative control space $-\mathcal{U}$, and apply all sampled control inputs to s_{near}^i for a small time increment $\delta t > 0$ to generate a set of possible new states S_{new}^i from any \mathcal{T}_i . Again, the nearest neighbor of s_{rand} , $s_{new}^i \in S_{new}^i$, is added to each \mathcal{T}_i . This approach is repeated until the number of iterations reaches a predefined limit. Finally, a path selection process is executed to find a set of “optimal” feasible paths based on specific optimization criteria, or a failure report is output with no feasible plan found.

One concern of RRT-based motion planning algorithm with backchaining is the efficient growth of the reversed RRTs toward the entry region. In order to grow the forest quickly toward the entry region, we apply a biased sampling strategy in CS_{free} , in which the state s_{rand} is sampled with a higher density in S_{goal} than elsewhere in the configuration

space. If s_{rand} collides with any obstacle, it is discarded and new states are sampled until one in CS_{free} is found.

The control inputs are sampled uniformly in the control space, using CONTROL_SAMPLING(), within a predefined range $[v_{min}, v_{max}] \times [\omega_{min}, \omega_{max}]$. The full Algorithm 1 is shown in Table I and Table II.

B. Paths Selection

If a target s_{goal}^i is reachable from the entry region, Algorithm 1 will output a set of feasible paths for this target, denoted by P_i . SELECT_PATHS() will select and output the “best” plan, $P^* = \{p_1^*, \dots, p_n^*\}$, for the given target set S_{goal} , where p_i^* is the “best” path for s_{goal}^i . With consideration of different optimization criteria, different path selection strategies can be applied.

1) *Plan with minimal twists*: While inserting a needle into the patient’s body, rotating the needle base may cause damages to the tissue. Because of the discretization of the needle’s kinematics, the more segments a feasible path contains, the more damage it will cause on the tissue because each segment includes a twist. In addition, uncertainties in needles’ motions will greatly affect the performance and accuracy of the insertion. Since the control of the needles is open loop, the more control inputs a feasible path has, the more uncertainties this path will involve. Let n_k be the number of control inputs corresponding to the k^{th} path in P_i , the p_i^* for each s_{goal}^i can be selected using the following minimal twists strategy,

$$p_i^* = \arg \min_{p_k \in P_i} (n_k).$$

This minimizes the lengths of paths, which is equivalent in our formulation to minimizing the number of twists.

2) *Plan with minimal insertion region*: Inserting multiple needles to reach multiple targets may require an entry region with a larger surface area. This will increase the complexity of the treatment as well as the trauma on the patient. As described above, Algorithm 1 considers this issue by exploring the connectivity between different targets so that possible plan may be found to reach multiple targets with only one needle. Besides that, the entry region can be further minimized based on the output path sets, P_i s. Let e_k^i be the corresponding entry point of the k ’th path, $p_i^k \in P_i$, for target s^i . The optimal plan P^* can be selected by using the following minimal entry region strategy,

$$P^* = \arg \min_{k,l} (\max_{i,j} d(e_k^i, e_l^j)),$$

where $d(e_k^i, e_l^j)$ is the Euclidean distance between e_k^i and e_l^j . This strategy minimizes the maximal distance between possible entry points of any two needles.

The proposed path selection strategies above optimize across the paths that were computed by the RRT approach. Due to the probabilistic nature of RRTs and the termination of the algorithm based on iteration count, the paths generated by the RRT represent only a subset of the feasible paths. Hence, the paths computed by the selection strategies are not guaranteed to be optimal.

VI. SIMULATION AND DISCUSSION

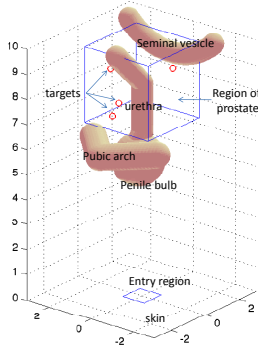


Fig. 4. The simulation setup: An approximated environment for multiple needle insertion.

A. Simulation setup

We implement the proposed RRT-Forest-based motion planning algorithm for multiple steerable needles insertion using an approximated 3D prostate environment. For our preliminary experiments, we use a set of spherical obstacles with different radii to approximate the real environment as shown in Fig. 4. The entire workspace is defined as a cuboid with coordinates $(-3, 3) \times (-3, 3) \times (0, 10)$. The region of interest around the prostate is defined as a cube of $3 \times 3 \times 3$ located at the top of the workspace. The feasible entry region is defined as a 2×2 square in the skin plane (X-Y plane). The assumption of spherical obstacles allows for fast collision detection. Interference of different needles is not considered until selecting the final optimal insertion plan. Simulations are run on a laptop with an AMD[®] Turion_{TM}64 2.1 GHz CPU, 4 GB memory, and the Microsoft[®] Windows Vista[™] operating system.

B. Simulation Results

The insertion task is to insert multiple needles from the 2×2 rectangular entry region centered at $(0, 0, 0)$ in the skin plane to reach 5 targets, which are randomly generated in

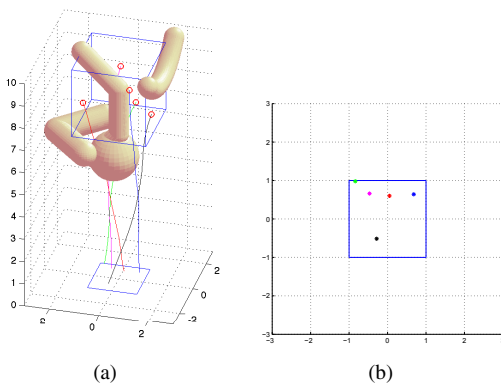


Fig. 5. A set of paths for simulation task 1: For all the 5 targets, only one path from each path set is selected with plan selection algorithm 1 to minimize twists of the path.

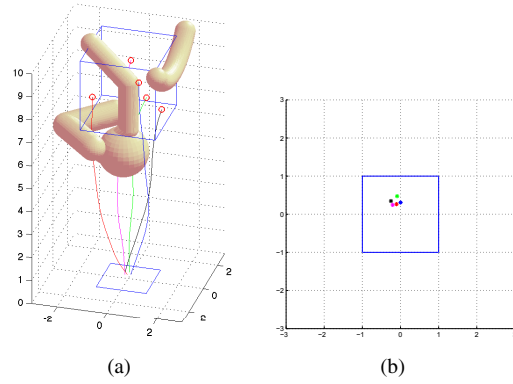


Fig. 6. A set of paths for simulation task 1: For all the 5 targets, only one path from each path set is selected with plan selection algorithm 2 to minimize region for insertion.

the region of interest. The range of the uniformly sampled control inputs are defined by insertion depth in the range of $[0.1, 0.5]$ and rotation angle in the range of $[0, 2\pi]$. We ran 5 trials, all of which successfully found feasible plans within 10000 iterations with an average computational time of 57573 seconds. Figure 7 shows the feasible plans found for all 5 targets in one trial.

We also compared Algorithm 1 to separately executing the SNST algorithm for each target. Using the SNST algorithm, feasible paths for each target were successfully found in 10000 iterations with a computational time of 87755 seconds. Algorithm 1 was 34% faster than executing the SNST algorithm multiple times. This is because it used every random sample to explore all 5 random trees, which is more efficient than separately applying the SNST algorithm.

Figure 5(a) shows the “optimal” plan selected using minimal twist strategy, and Figure 5(b) shows the corresponding entry points. Figure 6(a) shows the “optimal” plan selected using the minimal entry region strategy, and Figure 6(b) shows the corresponding entry points.

C. Discussion

The RRT-Forest-based algorithm efficiently explores the collision-free configuration space and its connectivity. With the minimal twist strategy, the entry points for all targets are selected in a way to minimize the path length to the corresponding target. However, this leads to a relatively larger entry region. With the minimal entry region strategy, the entry region is further minimized among all possible plans. As shown in Figure 6(b), the “optimal” plan for this trial can be reduced into a circular region of radius less than 0.3. This can significantly reduce the complexity of the treatment as well as the patient’s trauma.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an algorithm for motion planning of multiple bevel-tip steerable needles to reach multiple targets in 3D environments with obstacles. This algorithm is inspired by the RRT-based motion planning algorithm with backchaining for solving the feasible entry point planning

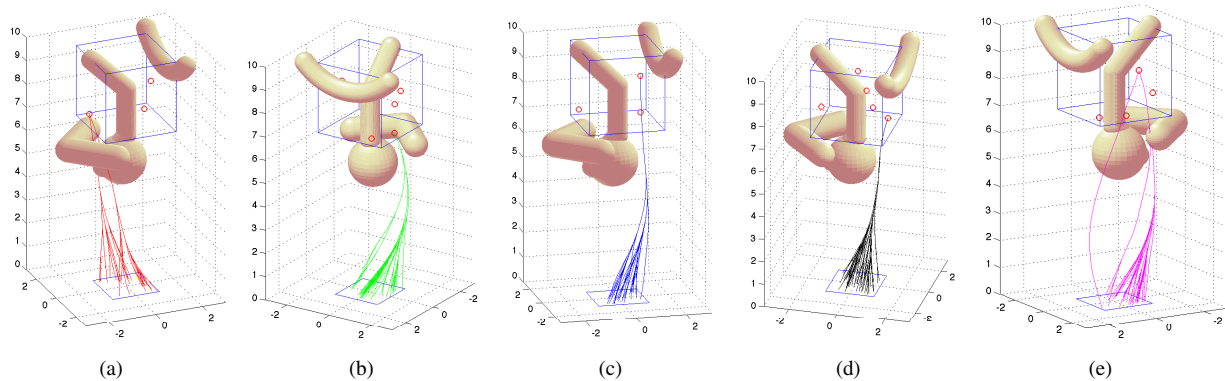


Fig. 7. Feasible paths found for one trial: For each of the 5 targets, a set of feasible paths is found. With any path in the path set, a steerable needle can be inserted from the single entry region to reach the target.

problem for a single needle [18]. The proposed algorithm builds an RRT forest data structure and provides a quick and more efficient exploration from all targets toward a single entry region to find feasible plans. We considered two path selection strategies: minimizing the number of needle twists and minimizing the size of the entry region. We conducted preliminary experiments in a 3D environment that approximates the prostate and surrounding obstacles.

In this paper, we only considered stiff environments with non-deformable spherical obstacles. In future work, we will explore motion planning for multiple steerable needles in deformable environments with more complex obstacles.

ACKNOWLEDGEMENT

The authors would like to thank Allison Okamura, Greg Chirikjian, Noah Cowan, and Kyle Reed at the Johns Hopkins University, and Vinutha Kallem at University of Pennsylvania for their valuable assistance to this work.

REFERENCES

- [1] R. Alterovitz, M. Branicky, and K. Goldberg, "Motion planning under uncertainty for image-guided medical needle steering," *Int. J. Robotics Research*, vol. 27, no. 11–12, pp. 1361–1374, Nov. 2008.
- [2] R. Alterovitz, K. Goldberg, and A. M. Okamura, "Planning for steerable bevel-tip needle insertion through 2D soft tissue with obstacles," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Apr. 2005, pp. 1652–1657.
- [3] R. Alterovitz, A. Lim, K. Goldberg, G. S. Chirikjian, and A. M. Okamura, "Steering flexible needles under Markov motion uncertainty," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Aug. 2005, pp. 120–125.
- [4] R. Alterovitz, T. Siméon, and K. Goldberg, "The Stochastic Motion Roadmap: A sampling framework for planning with Markov motion uncertainty," in *Robotics: Science and Systems III (Proc. RSS 2007)*, W. Burgard, O. Brock, and C. Stachniss, Eds. Cambridge, MA: MIT Press, 2008, pp. 246–253.
- [5] M. S. Branicky, M. M. Curtiss, J. Levine, and S. Morgan, "RRTs for nonlinear, discrete, and hybrid planning and control," in *Proceedings IEEE Conference Decision & Control*, 2003.
- [6] A. Cunha and J. Pouliot, "Dosimetric equivalence of non-standard brachytherapy needle patterns," *Medical Physics*, no. 6, pp. 2737–2737, 2008.
- [7] V. Duindam, R. Alterovitz, S. Sastry, and K. Goldberg, "Screw-based motion planning for bevel-tip flexible needles in 3d environments with obstacles," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2008, pp. 2483–2488.
- [8] V. Duindam, J. Xu, R. Alterovitz, S. Sastry, and K. Goldberg, "3d motion planning algorithms for steerable needles using inverse kinematics," in *Proc. Workshop on the Algorithmic Foundations of Robotics*, 2008.
- [9] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2000.
- [10] S. M. LaValle, *Rapidly-exploring random trees: a new tool for path planning*. TR 98-11, Department of Computer Science, Iowa State University, 1998.
- [11] —, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at <http://planning.cs.uiuc.edu/>.
- [12] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [13] M. D. O’Leary, C. Simone, T. Washio, K. Yoshinaka, and A. M. Okamura, "Robotic needle insertion: Effects of friction and needle geometry," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2003.
- [14] W. Park, J. S. Kim, Y. Zhou, N. J. Cowan, A. M. Okamura, and G. S. Chirikjian, "Diffusion-based motion planning for a nonholonomic flexible needle model," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2005, pp. 4611–4616.
- [15] S. Rodriguez, X. Tang, J. Lien, and N. Amato, "An obstacle-based rapidly-exploring random tree," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006, pp. 895–900.
- [16] R. J. Webster III, J. Memisevic, and A. M. Okamura, "Design consideration for robotic needle steering," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2005.
- [17] R. J. Webster III, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *Int. J. Robotics Research*, vol. 25, pp. 509–525, May 2006.
- [18] J. Xu, V. Duindam, R. Alterovitz, and K. Goldberg, "Motion planning for steerable needles in 3d environments with obstacles using rapidly-exploring random trees and backchaining," in *Proc. IEEE Int. Conf. Automation Science and Engineering (CASE)*, Aug. 2008, pp. 41–46.