Proceedings of the 8th World Congress on Mechanical, Chemical, and Material Engineering (MCM'22)
Prague, Czech Republic – July 31, 2022 - August 02, 2022
Paper No. ICMIE 124
DOI: 10.11159/icmie22.124

Path Planning with Modified RRT* Algorithm for Lung Biopsy

Yuexi Dong, Kunpeng Wang, Zheng Yang, Sai Cheong Fok, Han Wang
Sichuan University-Pittsburgh Institute
Sichuan University, Chengdu, China
chelseadong@stu.scu.edu.cn; kunpeng.wang@scu.edu.cn; zhengyang2018@scu.edu.cn; saicheong.fok@scupi.cn;
wanghan08@stu.scu.edu.cn

Abstract – Path planning plays a central role in robot-assisted percutaneous insertion. The main challenge of path planning exists in the motion constraints inherited from the geometry and mechanics of the needle, and the complex anatomic environment of human body. In nonholonomic planning, the classic Rapidly-Exploring Random Trees (RRT) algorithm may fail to provide a continuous and obstacle-avoidable path. To find a feasible path and minimize the damage on soft tissues based on a newly-introduced curvature-controllable steerable needle, we propose a method that utilizes RRT* and quadratic Bezier curve smoothing technique. RRT* with Bezier Curve Smoothing can generate a path composed of smooth piecewise planar curves with continuous connections. Comparisons are employed to show that our method generates shorter and less torturous paths with a higher success rate.

***Keywords*:** Path planning, Flexible needle, Rapidly-Exploring Random Trees* (RRT*), Bezier Curve Smoothing

1. Introduction

In the last decade, there is growing interest on the advancement of percutaneous lung biopsy using robotic systems. Issues that need to be addressed in this development include the interaction of the path planning with the steering of the flexible bevel-tip needle to mitigate the associated risk of damages to soft tissues, blood vessels, and vital organs. Among the many studies on path planning, the Rapidly-Exploring Random Trees (RRT) algorithm is one of the more popular approaches, that has been extensively improved and modified to generate reasonable paths meeting the motion constraints of steerable needles while avoiding known obstacles. For example, Xu et al. [1] proposed a backchaining method based on RRT to find an entry point by growing a path backward from the target point; while Patil and Alterovitz [2] developed a 3D strategy by combining RRT with a reachability-guided sampling heuristic to generate bounded-curvature needle trajectories. Similarly, Bernardes et al. [3] and Xiong et al. [4] presented an arc-based RRT planner in 2D, which was later improved by Patil et al. [5] to achieve a rapid replanning approach in 3D.

The path planning in the aforementioned approaches were developed based on the unicycle model, which is a commonly used nonholonomic kinematic representation of the bevel-tip needle [6]. The path of steerable needle is assumed to be composed of a series of standard arcs whose curvature is upper bounded by the natural maximum curvature determined by the properties of soft tissues and needles [7]. This assumption reduces tissue damages during needle steering. However, it is difficult to steer the needle along the planned path in practical applications due to the inhomogeneity and deformation of soft tissues as well as the incomplete transmissions of rotation from the needle base to the tip. To overcome this problem, a curvature-controllable steerable (CCS) bevel-tip needle was examined by Bui et al. [8]. Recently, Xu et al. [9] developed a fuzzy-logic controller to steer the CCS needle. With this development, the conventional constant-curvature constraint in RRT can be relaxed as the damage to tissues can be minimized through feedback controls of the needle. This simplifies the path planning to searching for an optimal non-arbitrary 3D path that is continuous and piecewise planar through the obstacles. The RRT* approach is ideal for this application. For example, Zhang et al. [10] applied the RRT* algorithm with artificial potential field to search for an optimal path with robust obstacle-avoidance capability in multilayer tissues.

This paper focuses on the path planning for lung biopsy. The proposed approach, which is based on the CSS needle and the RRT* algorithm, aims to alleviate tissue damages by generating a collision-free path with shorter path length as well as less twists. Continuity of the path is guaranteed by geometric-curve-based smoothing with quadratic Bezier curves [11, 12]. The performance of the proposed approach is examined by comparing the planned 2D path with the RRT approach in an environment with similar obstacles using different start and target points. Criteria for comparison include the average time to search for a valid path, the success rate, average path length, and average number of twists. The proposed approach with

Bezier curve smoothing is also compared with the RRT* by Zhang et al. [10] in a 3D environment. The result of the comparison indicates that the proposed approach can generate a smoother and shorter path.

2. Proposed Path Planning Algorithm

The path planning assumed that the kinematic model of the CCS bevel-tip needle is valid during trajectory steering. In conventional RRT planning, the path is a sequence of continuous arcs tangential to one another. The constant curvature of each arc must not exceed the maximum natural curvature. The modified RRT with local planner generating arc can search for several feasible paths, but the paths produced are often longer due to redundant twists. To address these issues, Karaman et al. [13] proposed the RRT*, which removes redundant edges so that the total path length is significantly reduced. This original RRT* needs to be modified for lung biopsy. The proposed changes include the following:

1) Goal bias is incorporated such that a goal node can be selected as a random node with a certain probability and an uncertainty range. This is to cater for the changes in tumor location due to lung motion, and could also lead to faster convergence [14].
2) The "Branching Angle" as illustrated in Figure 1 will be constrained within a user-defined range. This is to reduce tissue damages by ensuring that there are no steep or sharp turns along the created path.

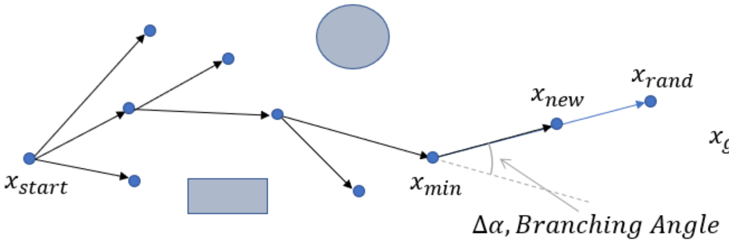


Fig. 1: Illustration of Branching Angle.

3) Bezier curve smoothing is applied to the searched path as illustrated in Figure 2. This is to generate a final path that is continuous and piecewise planar.

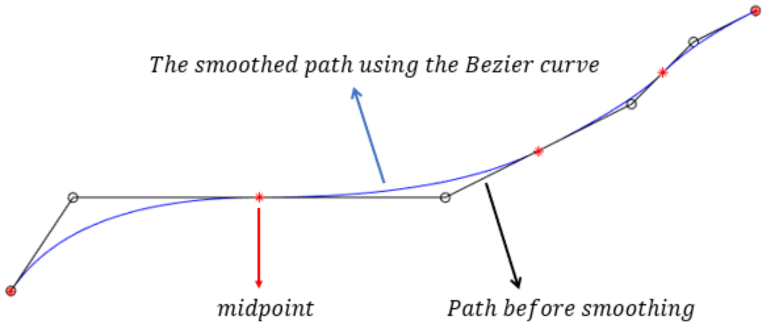
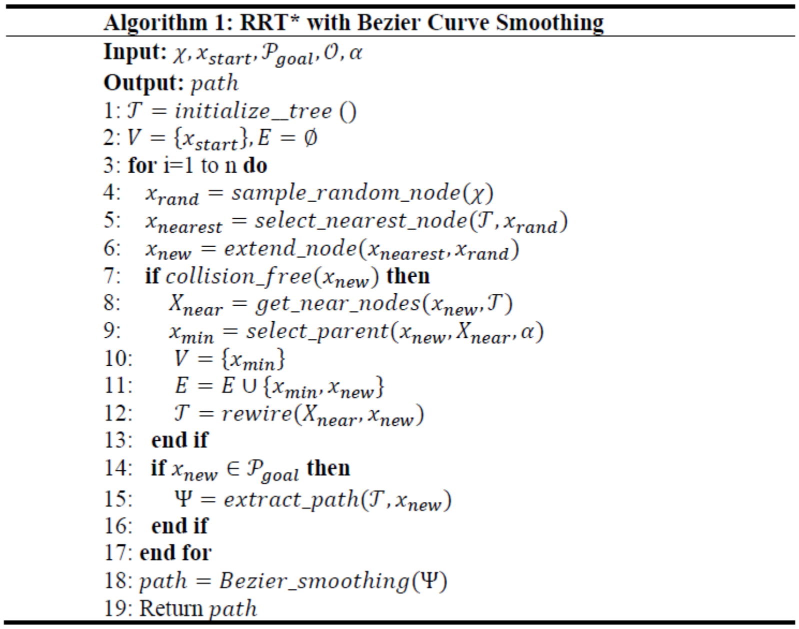


Fig.2: Post smoothing using quadratic Bezier curve.

Figure 3 shows the pseudo-code for the RRT* with the Bezier curve smoothing. The input to the algorithm is the configuration space, denoted by $\mathcal{X}$, the start configuration, x_{start}, the goal region, $\mathcal{P}_{goal}$, and the obstacle region, $\mathcal{O}$. The goal region, $\mathcal{P}_{goal}$, is defined as a spherical region with the goal configuration, x_{goal}, as the center. V and E represent the vertices and the edges of the search tree, respectively. The search tree is first initialized with V containing

x_{start} and E is set empty. The tree incrementally branches out by generating random nodes with a goal-biased sampling strategy. Without the goal bias, a completely random node, x_{rand}, is drawn from the free space based on a uniform distribution. Under a goal-biased sampling approach, the goal node, x_{goal}, is set as the random node with a certain probability probability within the goal region. The search then traverses every node on the tree to find the node, $x_{nearest}$, that is nearest nearest to x_{rand} based on Euclidean distance. If the distance between $x_{nearest}$ and x_{rand} is greater than one step length, a a new node, x_{new}, is generated at one step length in the direction prescribed by $x_{nearest}$ and x_{rand}. Otherwise, x_{rand} is set as x_{new}. Next, a collision check is conducted to ensure that x_{new} does not lie in $\mathcal{O}$. The function *get_near_nodes()* in Figure 3 allows a local search for the set of potential parent nodes x_{new} within the region defined by x_{new}. The parent node, x_{min}, is selected from x_{new} based on the criteria of minimum distance from the stating node and within the range of “Branching Angle”. Following the selection of the parent node, x_{min} and x_{new} are linked with a straight line. A rewiring of the tree is then performed. The above process is terminated when x_{near} is in $\mathcal{P}_{goal}$, with which the path Ψ is traced and extracted. Otherwise, the iteration is repeated until x_{new} is found to be in $\mathcal{P}_{goal}$.



Algorithm 1: RRT* with Bezier Curve Smoothing

```
Input: $\chi, x_{start}, \mathcal{P}_{goal}, \mathcal{O}, \alpha$
Output: $path$
$\mathcal{T} = initialize\_tree\ ()$
$V = \{x_{start}\}, E = \emptyset$
for i=1 to n do
   $x_{rand} = sample\_random\_node(\chi)$
   $x_{nearest} = select\_nearest\_node(\mathcal{T}, x_{rand})$
   $x_{new} = extend\_node(x_{nearest}, x_{rand})$
   if $collision\_free(x_{new})$ then
      $X_{near} = get\_near\_nodes(x_{new}, \mathcal{T})$
      $x_{min} = select\_parent(x_{new}, X_{near}, \alpha)$
      $V = \{x_{min}\}$
      $E = E \cup \{x_{min}, x_{new}\}$
      $\mathcal{T} = rewire(X_{near}, x_{new})$
   end if
   if $x_{new} \in \mathcal{P}_{goal}$ then
      $\Psi = extract\_path(\mathcal{T}, x_{new})$
   end if
end for
$path = Bezier\_smoothing(\Psi)$
Return $path$
```

Fig.3: Pseudo-code for main RRT* with Bezier Curve Smoothing.

The final stage of the proposed approach involves the smoothing of the straight-line segments using Bezier curve. Given $n + 1$ control points $\boldsymbol{P_0}$, $\boldsymbol{P_1}$, $\boldsymbol{P_2}$, ..., $\boldsymbol{P_n}$, the Bezier curve of degree n, is defined as

$$\boldsymbol{P}(t) = \sum_{i=0}^{n} B_{i,n}\,(t)\boldsymbol{P_i}, t \in [0,1], \tag{1}$$

where $B_{i,n}(t)$ is referred to as the Bezier basis functions (or Bernstein polynomials) with the following function

$$B_{i,n}(t) = \frac{n!}{i!\,(n-i)!}t^i(1-t)^{n-i}, i = 0, \dots, n, n \geq 1, t \in [0,1]. \tag{2}$$

Differentiating the parametric function (1) of the Bezier curve gives

$$\boldsymbol{P}'(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t)(\boldsymbol{P_{i+1}} - \boldsymbol{P_i}), t \in [0, 1]. \quad (3)$$

Setting $\boldsymbol{Q_0} = n(\boldsymbol{P_1} - \boldsymbol{P_0}),\ \boldsymbol{Q_1} = n(\boldsymbol{P_2} - \boldsymbol{P_1}), \dots, \boldsymbol{Q_{n-1}} = n(\boldsymbol{P_n} - \boldsymbol{P_{n-1}})$, the above can be rewritten as

$$\boldsymbol{P}'(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t)\boldsymbol{Q_i}, t \in [0, 1], \quad (4)$$

where $\boldsymbol{Q_i}, i = 0, \dots, n-1$, is the vector connecting consecutive control points as defined above.

Hence, the first order derivative of an n-degree Bezier curve is a Bezier curve of degree $n-1$. This characteristic of the Bezier curve allows the smoothing of jagged piecewise path. It follows from the equation (1) that the curve passing the first and last control points at $t = 0$ and $t = 1$, respectively. Equation (4) gives $\boldsymbol{P}'(0) = n(\boldsymbol{P_1} - \boldsymbol{P_0})$ and $\boldsymbol{P}'(1) = n(\boldsymbol{P_n} - \boldsymbol{P_{n-1}})$, indicating that the direction of the tangent vectors of the Bezier curve at the first and last control points are determined by the first and last line segments defined by $\boldsymbol{P_0P_1}$ and $\boldsymbol{P_{n-1}P_n}$, respectively. As a result, when two Bezier curves, for example, $\boldsymbol{C(t)}$ and $\boldsymbol{D(t)}$, defined by $m+1$ control points, are connected at one common point, G^1 continuity follows naturally since $\boldsymbol{C_m} = \boldsymbol{D_0}$ is the common point of intersection and directions defined by $\boldsymbol{C_{m-1}C_m}$ and $\boldsymbol{D_0D_1}$ are equi-directional. To achieve C^1 continuity, $\boldsymbol{C_{m-1}C_m}$ and $\boldsymbol{D_0D_1}$ must have the same magnitude [15].

In our path planning problem, we consider a smooth path evolving without sharp turns, not only because of the convenience in the practical operation, but also from the perspective of minimizing the tissue damage. The smoothing process involves finding the midpoints from the second edge to the penultimate edge of the path Ψ as shown in Figure 2. The Bezier curves are established via connecting the consecutive midpoints except at the initial and last point, at which we connect to the nearest midpoint. Each piece is a quadratic curve lying in a plane since we have three non-colinear control points. Consequently, G^1 and C^1 continuities are implied from connecting the consecutive Bezier curves at the common points of intersection, at which the connecting vectors, e.g., $\boldsymbol{C_{m-1}C_m}$ and $\boldsymbol{D_0D_1}$, at each respective Bezier curve are identical. After the smoothing treatment, a new path is generated following the trajectory of the Bezier curves joining the start and end points.

3. Results

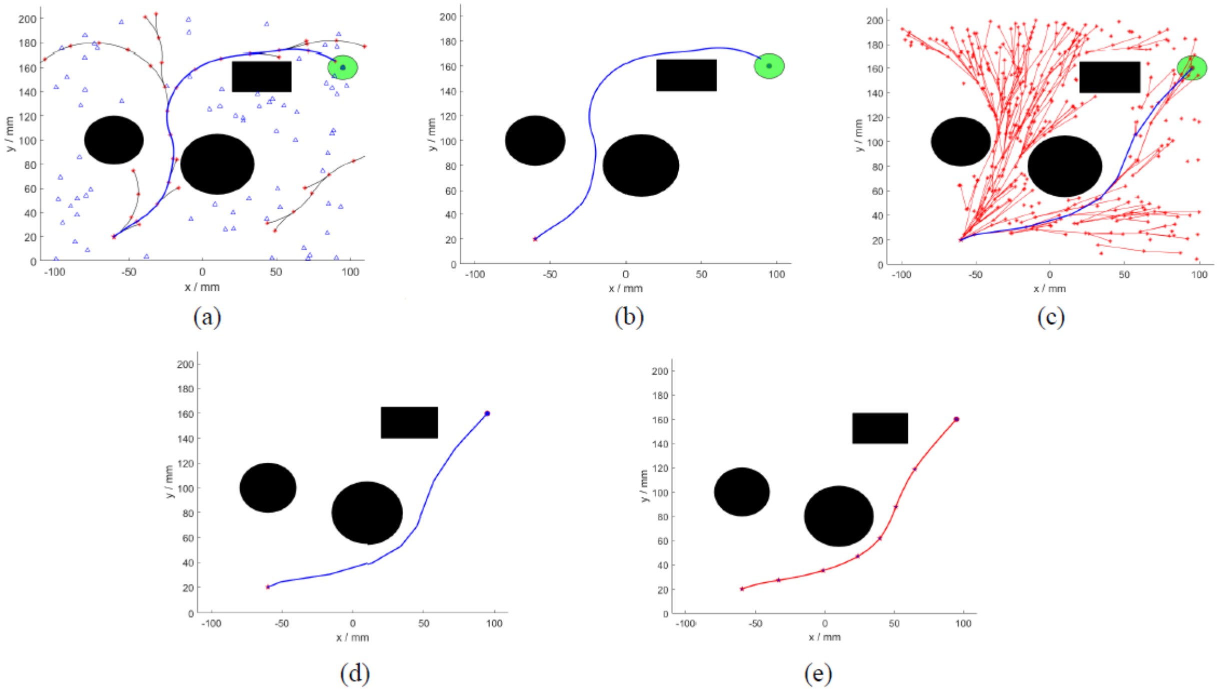
3.1. 2D Comparison of RRT and Proposed Algorithm

The proposed approach is compared with the RRT algorithm based on simulation using a 2D environment with random set-up obstacles. The configuration space is chosen as $\mathcal{X} = [-100, 100] \times [0, 200]$. Two different settings of start configuration, x_{start}, and goal configuration, x_{goal}, are shown in Table 1.

Table 1: Start and goal settings for simulation.

Setting	x_{start}	x_{goal}
1	$(-60, 20)$	$(95, 160)$
2	$(80, 20)$	$(-90, 160)$

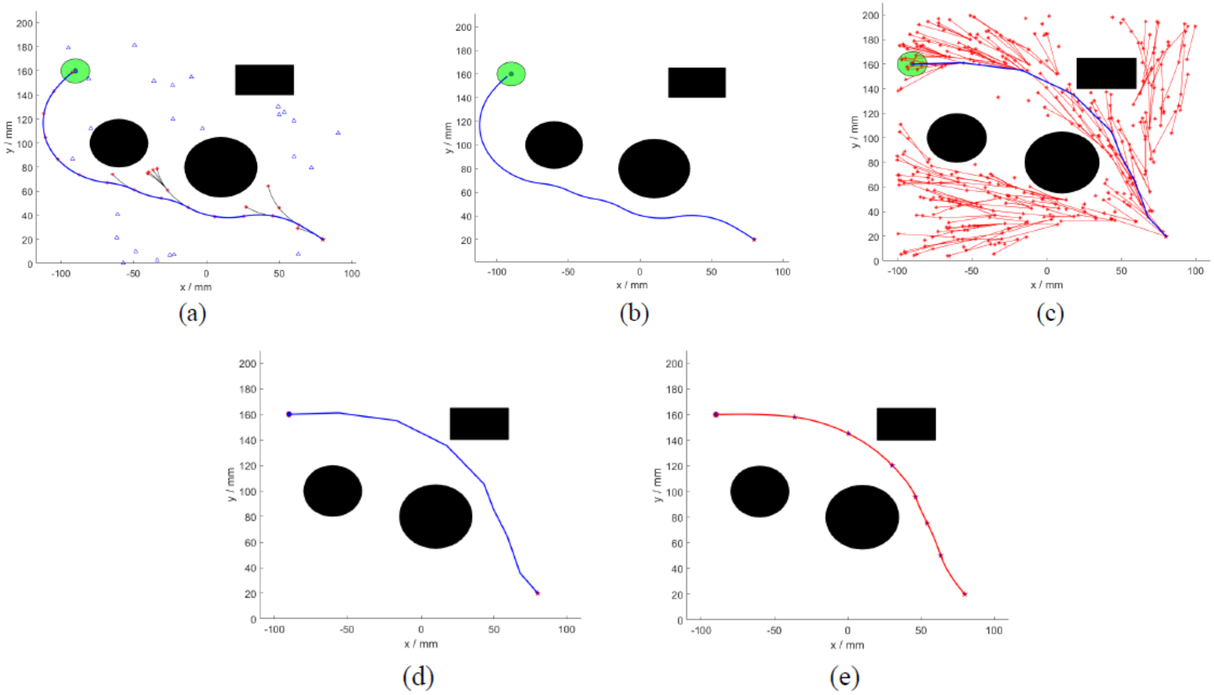
Samples of the search tree and final paths are demonstrated as shown in Figures 4 and 5. The green circles in Figures 4a, 4b, 4c, 5a, 5b. and 5c represent the corresponding goal region. We use a rectangle and circles as the obstacles. In each setting, the final paths can be slightly different due to the generation of the random nodes in the search space of both RRT and RRT* with Bezier Curve Smoothing. In general, the proposed algorithm gives less variation in the final paths due to the search criteria of the shortest path with constrained “Branching Angles”. The proposed approach also always gives a solution in each run. Note this is not the case with the RRT algorithm. In certain execution, the RRT search cannot find a solution. This is mainly due to the constraint of constant curvature. Besides the success rates, the time to search for a successful path also differs in the two approaches.



(a) (b) (c)

(d) (e)

Fig.4: Comparison of RRT and RRT* with Bezier Curve Smoothing in setting 1. (a) Search trees of RRT. (b) Final path of RRT. (c) Search trees of RRT* with Bezier Curve Smoothing. (d) Path of RRT* before smoothing. (e) Final path of RRT* with Bezier Curve Smoothing.



(a) (b) (c)

(d) (e)

Fig.5: Comparison of RRT and RRT* with Bezier Curve Smoothing in setting 2. (a) Search trees of RRT. (b) Final path of RRT. (c) Search trees of RRT* with Bezier Curve Smoothing. (d) Path of RRT* before smoothing. (e) Final path of RRT* with Bezier Curve Smoothing.

To compare the performances of the two approaches, 30 simulations for each setting were conducted. The average search time to reach a successful solution, the success rate, the average total length, and the average number of twists are compared. Here, the number of twists corresponds to how many times the needle needs to reorientate. The results of the comparison are shown in Table 2. Based on results in different settings of start and goal configuration, our proposed method outperforms RRT and on average computes a shorter and less torturous path. However, the computational time of RRT* with Bezier Curve Smoothing is longer than RRT. This is mainly due to the angle limitation during tree expansion as well as the parametrized smoothing with the Bezier curve. Physicians can strike a balance between a shorter and smoother path with slight extension of computation time, and a longer and torturous path. In practical scenario where rehabilitation of tissue is difficult and tissue damage is considered fatal, path planned by our proposed method would be preferable.

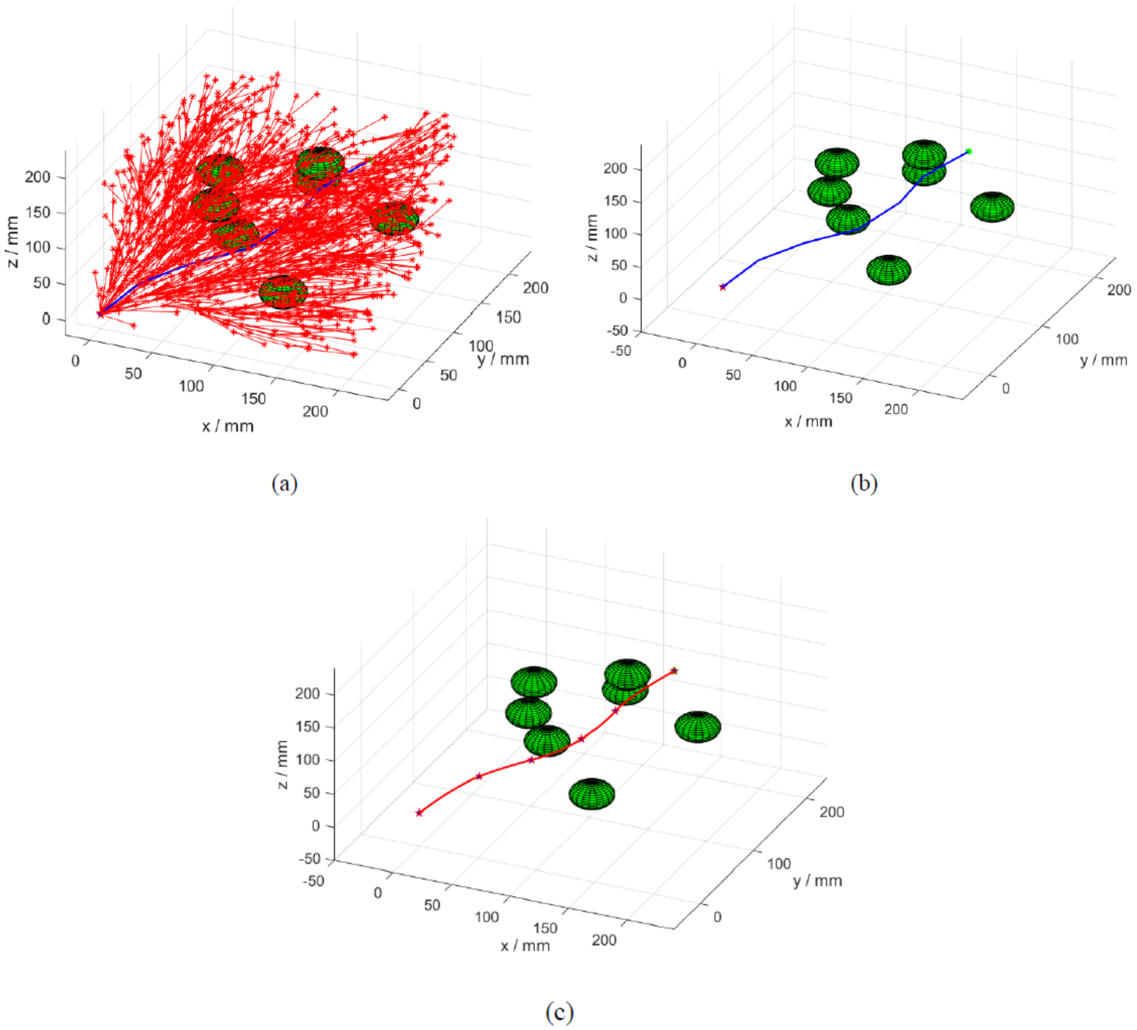
Table 2: Comparisons of simulation results of RRT and RRT* with Bezier Curve Smoothing by four evaluation metrics.

	Algorithm	The average run time (s)	The success rate	The average total length (mm)	The average number of twists
Setting 1	RRT	5.447	71.0%	259.09	13
	RRT* with Bezier Curve Smoothing	13.480	100.0%	222.59	7
Setting 2	RRT	6.126	86.7%	266.15	14
	RRT* with Bezier Curve Smoothing	12.570	100.0%	234.99	8

3.2. 3D Comparison of RRT* [10] and Proposed Algorithm

The proposed approach is compared with the RRT* algorithm by Zhang et al. [10] based on simulation using a similar 3D environment with known obstacles. The configuration space is set as $\mathcal{X} = [0,200] \times [0,200] \times [0,200]$. A representative simulation result for the proposed algorithm is shown in Figure 6. The start configuration, x_{start}, is the origin $(0,0,0)$ and goal configuration, x_{goal}, is $(150,150,150)$. Obstacles are represented as spheres. Figure 6a shows the search tree in the proposed algorithm and Figure 6b shows the final path before smoothing. Figure 6c shows the final path after smoothing.

As we do not have the precise coordinates and setups of the obstacles in Zhang et al, certain variations may exist. The final path length obtained by Zhang et al. is $291\,mm$, while the path produced by our proposed method is $268\,mm$. Regardless of the variations due to the uncertain coordinates, the shorter path obtained in the proposed algorithm is because of the relaxed curvature constraint in the kinematic model based on the CCS needle. The CCS needle allows us to construct a path with smoothly-changing curvature rather than one with piecewise constant curvature. The curvature obtained in the proposed algorithm is comparable with that of Zhang et al, which was set as $0.02\,mm^{-1}$. According to Bui et al. [8], the maximum curvatures of the CCS needle were measured to be $0.008\,mm^{-1}$ and $0.0139\,mm^{-1}$ in the experiment using $6\,wt\%$ and $10\,wt\%$ vitro gelatin phantom, respectively. In ex vivo cow liver, the tested maximum curvature is $0.0038\,mm^{-1}$. In our simulations, the average minimum curvature of the generated paths is $0.0016\,mm^{-1}$while the average maximum curvature is $0.0158\,mm^{-1}$. Based on these data, the application of the proposed algorithm in practical application is promising.



(a)

(b)

(c)

Fig.6: Comparison of the result of RRT* Bezier Curve Smoothing in 3D environment with Zhang et al. (a) Search trees of RRT* with Bezier Curve Smoothing. (b) Path of RRT* before smoothing. (c) Final path of RRT* with Bezier Curve Smoothing.

4. Conclusion

In this paper, we introduce a path planning algorithm, based on the CCS bevel-tip needle, to search and generate a smooth, continuous trajectory consisted of multiple pieces of planar curves for lung biopsy. The proposed algorithm is obtained by incorporating the RRT* path planning algorithm with quadratic Bezier Curve Smoothing. The performances of the proposed algorithm were compared in 2D simulations with the RRT algorithm and in 3D with the achieved results by Zhang et al. [10].

Findings based on the simulation results suggest that, RRT* with Bezier Curve Smoothing is promising and able to reach the goal point with higher success rate as well as shorter and smoother paths. The curvature of generated path is comparable with the experimental result using CCS bevel-tip needle in Bui et al. [8]. Nevertheless, there is still a lot of work to be done to further improve the proposed algorithm before it can be clinically tested. Future work includes minimization of the curvature deviation to facilitate the regulation of the steering needle, along with appropriate control strategies. More extensive simulation studies with experimental verification will need to be performed in inhomogeneous soft tissues.

Acknowledgements

This work was supported by the Young Scientist Fund of the Provincial Natural Science Foundation of Sichuan Province under Grant No.2022NSFSC1855.

References

[1] 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," 2008 IEEE International Conference on Automation Science and Engineering, 2008, pp. 41-46.

[2] S. Patil and R. Alterovitz, "Interactive motion planning for steerable needles in 3D environments with obstacles," 2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics, 2010, pp. 893-899.

[3] M. C. Bernardes, B. V. Adorno, P. Poignet, N. Zemiti and G. A. Borges, "Adaptive path planning for steerable needles using duty-cycling," 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, pp. 2545-2550.

[4] J. Xiong, X. Li, Y. Gan and Z. Xia, "Path planning for flexible needle insertion system based on Improved Rapidly-Exploring Random Tree algorithm," 2015 IEEE International Conference on Information and Automation, 2015, pp. 1545-1550.

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

[6] 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 (IJRR), vol. 25, no. 5-6, pp. 509–525, May 2006.

[7] D. S. Minhas, J. A. Engh, M. M. Fenske, and C. N. Riviere, "Modeling of needle steering via duty-cycled spinning," in Proc. Int. Conf. IEEE Eng. Med. Biol. Soc., 2007, pp. 2756–2759.

[8] V. K. Bui, S. Park, J. O. Park and S. Y. Ko, "A novel curvature-controllable steerable needle for percutaneous intervention, " Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, vol. 230, no. 8, pp. 727-738, 2016.

[9] B. Xu and S.Y. Ko, "3D Feedback control using fuzzy logic for a curvature-controllable steerable bevel-tip needle, " Mechatronics, vol. 68, pp. 102368, 2020.

[10] Y. Zhang, Z. Ju, H. Zhang and Z. Qi, "3-D Path planning using improved RRT* algorithm for robot-assisted flexible needle insertion in multilayer tissues," in IEEE Canadian Journal of Electrical and Computer Engineering, vol. 45, no. 1, pp. 50-62, winter 2022.

[11] I. Škrjanc and G. Klančar, "Optimal cooperative collision avoidance between multiple robots based on Bernstein-Bézier curves, " Robotics and Autonomous Systems, vol. 58, pp. 1–9, 2010.

[12] Y. Ho and J. Liu, "Collision-free curvature-bounded smooth path planning using composite Bezier curve based on Voronoi diagram," 2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA), 2009, pp. 463-468.

[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," Int. J. Robot. Res., vol. 30, no. 7, pp. 846–894, 2011.

[14] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning, " The International Journal of Robotics Research (IJRR), vol. 20, no. 5, pp. 378-400, 2001.

[15] C. K. Shene. (2011, May 4). Introduction to Computing with Geometry Notes. [Online]. Available: https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/