6-8-2016

3D Dynamic Motion Planning for Robot-Assisted Cannula Flexible Needle Insertion into Soft Tissue

Yanjiang Zhao
Harbin University of Science and Technology

Wenqiang Wu
Harbin University of Science and Technology

Yongde Zhang
Harbin University of Science and Technology

Ruixue Wang
Harbin University of Science and Technology

Jingchun Peng
Harbin University of Science and Technology

Recommended Citation

Authors

Yanjiang Zhao, Wenqiang Wu, Yongde Zhang, Ruixue Wang, Jingchun Peng, and Yan Yu

OPEN ACCESS
ARTICLE

3D Dynamic Motion Planning for Robot-assisted Cannula Flexible Needle Insertion into Soft Tissue

Regular Paper

Yan-Jiang Zhao[1]*, Wen-Qiang Wu[1], Yong-De Zhang[1], Rui-Xue Wang[1], Jing-Chun Peng[1] and Yan Yu[2]

1 Intelligent Machine Institute, Harbin University of Science and Technology, Harbin, China
2 Department of Radiation Oncology, Thomas Jefferson University, Philadelphia, USA
*Corresponding author(s) E-mail: zhaoyj@hrbust.edu.cn

Abstract

In robot-assisted needle-based medical procedures, insertion motion planning is a crucial aspect. 3D dynamic motion planning for a cannula flexible needle is challenging with regard to the nonholonomic motion of the needle tip, the presence of anatomic obstacles or sensitive organs in the needle path, as well as uncertainties due to the dynamic environment caused by the movements and deformations of the organs. The kinematics of the cannula flexible needle is calculated in this paper. Based on a rapid and robust static motion planning algorithm, referred to as greedy heuristic and reachability-guided rapidly-exploring random trees, a 3D dynamic motion planner is developed by using replanning. Aiming at the large detour problem, the convergence problem and the accuracy problem that replanning encounters, three novel strategies are proposed and integrated into the conventional replanning algorithm. Comparisons are made between algorithms with and without the strategies to verify their validity. Simulations showed that the proposed algorithm can overcome the above-noted problems to realize real-time replanning in a 3D dynamic environment, which is appropriate for intraoperative planning.

1. Introduction

In minimally invasive surgeries, needle insertion is likely one of the most popular procedures, applied in tissue biopsies and radioactive brachytherapies for cancers. However, targeting is a challenge when targets are surrounded by anatomic obstacles or sensitive organs that must be avoided. Traditional rigid needles are not useful for this task. Therefore, a bevel tip flexible needle is proposed for overcoming this problem [1]. The conventional flexible needle is supposed to be flexible to a degree that it can bend inside tissue, due to lateral force being applied on the bevel tip at insertion. However, it still has some drawbacks: firstly, the curvature of the trajectory for a needle is supposed to be fixed in a tissue and cannot rectify its path once it has deviated. Secondly, it is difficult to precisely control the orientation of the bevel tip at

rotation, due to the torsional friction between the needle shaft and tissue. To overcome the first drawback, Minhas and Majewicz utilized the duty-cycling of the spinning motor to generate a series of curvatures [2, 3]. Datla et al. proposed an active, flexible needle that takes advantage of the characteristics of shape memory alloys (SMA), which can also generate different curvatures by supplying different electric currents to the SMA actuators [4-6]. However, the second drawback remains in place. We have been developing a cannula flexible needle, composed of a flexible cannula and a flexible bevel-tip stylet, as shown in Fig.1. It can overcome both drawbacks of the conventional bevel tip flexible needle mentioned above. Firstly, it can generate a series of curvatures of trajectories by the different the length d of the stylet out of the cannula. Secondly, it can improve the rotation precision of the bevel tip, because the cannula separates the stylet and the tissue, thereby reducing torsional friction.
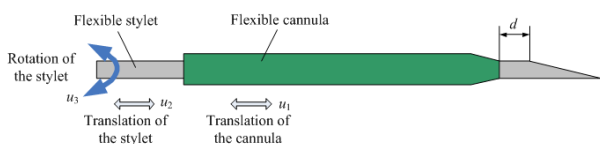

Figure 1. Schematic of a cannula flexible needle

Motion planning of the cannula flexible needle in the soft tissue is challenging due to the nonholonomic motion of the needle and the presence of anatomic obstacles and sensitive organs [7]. It is even complicated in a dynamic environment, with uncertainties present due to errors in needle tip positioning and needle modelling, the inhomogeneity and deformation of tissue and the physiological movement of organs [8]. All of these disturbances and movements may cause the needle to deviate from its intended path, which can have fatal consequences.

Motion planning for the conventional bevel tip flexible needle has been extensively studied using different approaches, which can be used as references for the cannula flexible needle. Duindam et al. proposed motion planning using discretization of the control space in a 3D environment with obstacles and formulated the motion planning problem as a nonlinear optimization problem [7]. However, this planning is applied for a static environment. Alterovitz et al. depicted the motion planning problem as a Markov decision process, using a discretization of the state space to maximize the probability of successfully reaching the target in a 2D environment [8]. Park et al. proposed a path-of-probability algorithm to optimize the paths by computing the probability density function [9]. Both of the above-noted approaches considered the uncertainty regarding the needle's response to control using dynamic programming; however, these approaches are applied for a quasi-dynamic environment, i.e., there is no actual disturbance or movement of the environment and the uncertainty is not actually formulated or updated to the

planning in question. All of the above approaches adopted a mathematical computation (MC) method, which configures the problem as an optimization problem with an objective function and then computes the numerical optimal solution. This method generally has a computational expense and may suffer from convergence; therefore, such methods are generally used for preoperative planning, but are not appropriate for intraoperative planning.

Another method for considering uncertainty, particularly in the case of tissue deformation is the finite element mesh (FEM) method. Alterovitz et al. used a FEM to compute soft tissue deformations and combined it with a MC method to find a locally optimal trajectory [10]. Patil et al. proposed motion planning for highly deformable environments, using the FEM method combined with a sampling-based algorithm [11]. However, the efficiency of the FEM method depends significantly on how accurately the mesh represents the real tissue. Moreover, both of these studies did not consider many other uncertainties caused by the surrounding environment. Moreover, the FEM method is also time-consuming. Therefore, it is generally used in preoperative motion planning. However, if we consider real-time dynamic motion planning, it may not be appropriate.

A third important method for flexible needle motion planning is a sampling-based method such as the probabilistic roadmaps (PRM) or the rapidly-exploring random trees (RRT) method. Some motion planning has been developed based on PRM in a static environment or a quasi-dynamic environment [12, 13]. Ever since Xu et al. first applied a RRT-based method to search for valid needle paths in a 3D environment with obstacles, the RRT algorithm has been commonly used in flexible needle motion planning [14]. However, these studies were only considered in a static environment. Patil et al. utilized improved RRTs called reachability-guided RRTs (RG-RRTs) and achieved orders of magnitude speed-ups compared to previous approaches, and relaxed the constraint of constant-curvature needle trajectories by relying on duty-cycling to realize bounded-curvature needle trajectories. This enabled the RRT method to be appropriate for dynamic intraoperative motion planning [15]. Caborni et al. proposed risk-based path planning for a steerable flexible probe based on the RG-RRTs [16]. In terms of path replanning, Patil et al. proposed a rapid replanning algorithm based on RG-RRTs and conducted experimental evaluations to show its validity [17]. Vrooijink et al. also developed a closed-loop replanning algorithm based on the RG-RRTs for a 3D non-static environment using 2D ultrasound images [18]. Both of these studies considered the uncertainty arising from tissue deformations, actuation errors and noisy sensing; however, they did not consider the significant displacement of targets and obstacles caused by physiological movement, which may cause the needle to fatally deviate from the planned path. Bernardes et al. presented a fast intraoperative replanning algorithm that considers disturbances including the movements of

obstacles and a target based on RRTs in both 2D and 3D environments; however, target movement was limited in 2D, despite having the potential to move in 3D [19, 20].

In summary, the advantage of the RRT method is that it is very fast (in milliseconds), easy to implement and probabilistic to complete, which is suitable for intraoperative planning. However, for dynamic replanning, all replanning algorithms may suffer from the large detour problem, convergence problem and/or accuracy problem. The large detour problem may be caused by the unpredicted movement of the obstacles and/or target; the convergence problem may be caused by nonholonomic motion and/or the minimum bending radius of the needle in the tissue, and the potential for the path adjustment becomes weaker when the needle gets close to the target; the accuracy problem may be caused by the termination conditions when the convergence problem occurs.

In this paper, we firstly calculate the kinematics of the cannula flexible needle, then introduce a fast motion planning algorithm based on RRT for a static environment (preoperative operation) for the cannula flexible needle in our previous work. Based on the proposed static motion planning algorithm, we propose dynamic (intraoperative operation) motion planning using replanning, taking into account any uncertainty caused by errors in needle tip positioning, the approximation of needle modelling, the inhomogeneity and deformation of tissue and physiological movement of the organs. Aiming at the large detour problem, the convergence problem and the accuracy problem that replanning encounters, we propose three novel strategies integrated into the intraoperative path replanning algorithm for overcoming these problems.

2. Kinematic Analysis of the Cannula Flexible Needle

Different from conventional bevel tip needles (with two DOFs: insertion and rotation) [1], the proposed cannula flexible needle has three DOFs: two translations u_1, u_2 for the cannula and the stylet, respectively, and rotation u_3 for the stylet,u_1, u_2 and u_3 can be input separately or simultaneously (see Fig.1).

2.1 Trajectory form analysis

We assume that the cannula flexible needle is pliably flexible and torsionally stiff, i.e., the needle shaft follows the needle tip, performing an approximate circular arc in the tissue when bending, while needle tip rotates at the same angle as the base [1, 7]. Similar to the conventional bevel tip flexible needles, it will be bent against the bevelled side when inserted into the tissue, due to the lateral force. The rotation of the stylet decides the orientation of the needle bending, so that the needle can achieve various paths in 3D by combining the three inputs.

1. Arc-based path: the radius of the needle path can vary according to the length d of the stylet which remains

out of the cannula (see Fig.1). The more the stylet is able to get out of the cannula, the smaller radius the trajectory will receive. Thus, the relative velocity between u_1 and u_2 changes the radius of the path, and u_3 changes the orientation of the bending. As u_1 and u_2 are generally input simultaneously and with the same velocity, we can denote them as u_{12}. If we input u_{12} and u_3 in a time-sharing way, the needle will generate an arc-based path (see Fig.2a). This method does not generate all radii of the path. The path will be in a range with the minimum bending radius r_{min} and the maximum bending radius r_{max} (as shown in Fig.3a). Both r_{min} and r_{max} are not only related to d, but also to the mechanical properties of the needle and the tissue. For the paths with a bending radius larger than r_{max}, we can also apply the duty-cycling method [2, 3]. By using a combination of both methods, the control expense is saved to some extent. The entire workspace is shown in Fig.3b.

2. Helix-based path: if we input u_{12} and u_3 simultaneously, and the speed of u_3 is much slower than that of u_{12}, the needle will generate a helix-based path (see Fig.2b).

3. Linear path: if we input u_{12} and u_3 simultaneously, however, the speed of u_3 will be much faster than that of u_{12}, and the needle will generate a linear path (see Fig.2c).

a) Arc-based path.   b) Helix-based path.

c) Linear path.

Figure 2. Forms of path

a) Workspace without duty cycling.   b) Workspace with duty cycling.
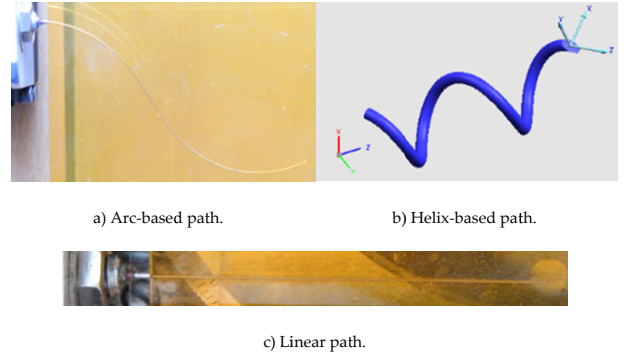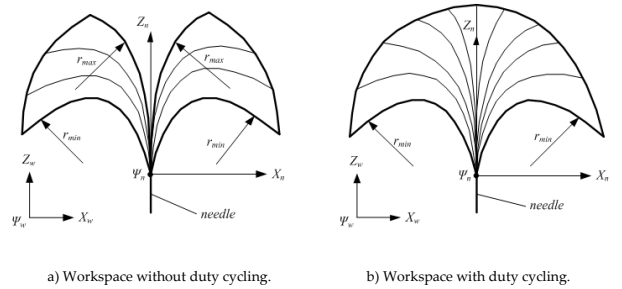
Figure 3. Workspace of the needle

In this paper, we adopt the arc-based and linear paths rather than the helix-based path, because they are easier to

control. The helix-based path, in contrast, is difficult to formulate and control and may also cause more trauma to biological organs, because its winding approach may lengthen the path.

2.2 Kinematic model

The kinematic model of the cannula flexible needle is formulated as shown in Fig.4. In the world frame Ψ_w, the configuration of the needle tip frame Ψ_n can be described compactly by a 4 x 4 homogeneous transformation matrix:

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

where $R_{wn} \in SO(3)$ and $p_{wn} \in \mathbb{R}^3$ are the rotation matrix and the position of frame Ψ_n relative to the frame Ψ_w, respectively.
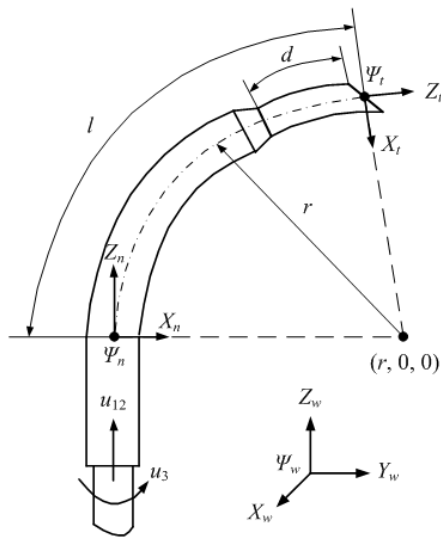

Figure 4. Kinematic model of the cannula flexible needle

In frame Ψ_n, the instantaneous linear velocity is $v = [0 \quad 0 \quad u_{12}]^T$, angular velocity is $\omega = [u_{12}/r_i \quad 0 \quad u_3]^T$, where r is the radius of the path and the twist $\xi \in se(3)$ is formulated as:

$$\hat{\xi} = \begin{bmatrix} v \\ \omega \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -u_3 & 0 & 0 \\ u_3 & 0 & -u_{12}/r & 0 \\ 0 & u_{12}/r & 0 & u_{12} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

where $\wedge$ is the wedge operator for forming a matrix in $se(3)$ out of a given vector in $\mathbb{R}^6$.

Then, the homogeneous transformation matrix can be formulated in exponential form as:

$$g_{wn}(t) = g_{wn}(0)\exp(\hat{\xi}t) \tag{3}$$

where $g_{wn}(0)$ is the initial configuration, which is the configuration of the needle (frame Ψ_n) in frame Ψ_w before insertion and t is the execution time. Additional details can be found in [1].

However, different to other existing kinematics of the flexible needle [1, 7, 9, 11-20], because the planning algorithm considers the insertion pose of the needle, we have to add an insertion configuration to the kinematics:

$$g_{wn}(t) = g_{wn}(0)g_{in}\exp(\hat{\xi}t) \tag{4}$$

where g_{in} is the insertion configuration.

The entire path can be discretized by a few segments. Let $g_i(t_i) = \exp(\hat{\xi}t_i)$ represent the transformation matrix of the ith segment, then the forward kinematics after N segments of execution is formulated as:

$$g_{wn}(T) = g_{wn}(0)g_{in}\prod_{i=1}^{N}g_i(t_i) \tag{5}$$

where t_i is the execution time of the ith segment, T is the total execution time of the whole path, $T = \sum_{i=1}^{n} t_i$; $g_{wn}(0)$ is the initial configuration and g_{in} is the insertion configuration.

To simplify the calculation, we make the Ψ_n frame coincide with the Ψ_w frame initially (see Fig.5). Thus, $g_{wn}(0) = I_4$, where I_4 is a unit matrix with 4 x 4. The insertion pose can be regarded as frame Ψ_n rotating α_0 around axis Z_n, then rotating β_0 around axis Y_n; hence, configurations for the insertion configuration can be obtained by:

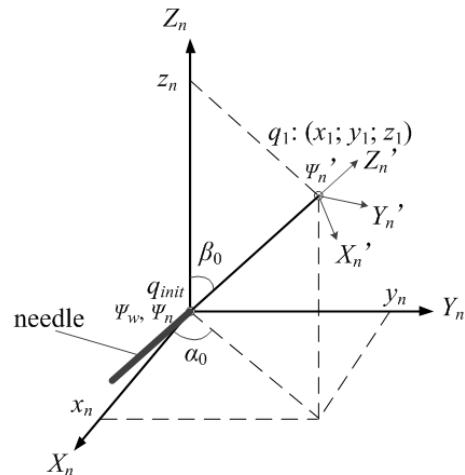$$g_{in} = Rot(Z_n, \alpha_0)Rot(Y_n, \beta_0) \tag{6}$$

Figure 5. Insertion configuration of the needle

3. Static Motion Planning Algorithm

We utilized the RRT method for the static motion planning, combined with the greedy heuristic method and reachable guided strategies, known as greedy heuristic and reachability guided rapidly-exploring random trees (GHRG-RRTs). We adopted variable but bounded curvatures for the needle paths and also took account of linear segments and relaxation of insertion orientations where the trajectories were concerned. The superiority of this algorithm is attested to in our previous work [21, 22].

3.1 Outline of GHRG-RRTs algorithm

The outline of the GHRG-RRTs algorithm is shown in Algorithm 1. Further details can be obtained in [22].

Algorithm 1: GHRG-RRTs (q_{init}, q_{goal}, Q).

1: $T \leftarrow$ InitTree(q_{init}); $P \leftarrow$ InitPath(q_{init});

2: **if** LinearCheck(q_{init}, q_{goal}, Q)

3: $U \leftarrow$ SolveLine(q_{init}, q_{goal})   ⎤

4: $P \leftarrow$ AchivePath(T, U, q_{goal})   ⎬ Get a linear path

5: **end if**   ⎦

6: $U \leftarrow$ SolveCurve($T.g_{init}$, q_{goal})   ⎤

7: **if** $U.r \geq r_{min}$ & CollisionFree(U, Q)   ⎬ Get a curve path

8: $P \leftarrow$ AchivePath(T, U, q_{goal})   ⎪

9: **end if**   ⎦

10: **while** ($n <$ max_path) & ($i<$ max_iteration)

11: $q_{rand} \leftarrow$ RandomNode(); flag $\leftarrow$ false

12: **if** LinearCheck(q_{init}, q_{rand})   ⎤

13: $U \leftarrow$ SolveLine(q_{init}, q_{rand})   ⎬ Get a linear tree

14: $T \leftarrow$ ExtendTree (T, U, q_{rand})   ⎦

15: $U \leftarrow$ SolveCurve($T.g_{rand}$, q_{goal})   ⎤

16: **if** $U.r \geq r_{min}$ & CollisionFree(U, Q)   ⎪

17: $P \leftarrow$ GetPath (T, U, q_{goal}); flag $\leftarrow$ true   ⎬ Get a path

18: **end if**   ⎦

19: **end if**

20: $U \leftarrow$ SolveCurve($T.g_{init}$, q_{rand})   ⎤

21: **if** $U.r \geq r_{min}$ & CollisionFree(U, Q)   ⎪

22: $T \leftarrow$ ExtendTree(T, U, q_{rand})   ⎬ Get a curve tree

23: **end if**   ⎦

24: $U \leftarrow$ SolveCurve($T.g_{rand}$, q_{goal})   ⎤

25: **if** $U.r \geq r_{min}$ & CollisionFree(U, Q)   ⎪

26: $P \leftarrow$ GetPath(T, U, q_{goal}); flag $\leftarrow$ true   ⎬ Get a path

27: **end if**   ⎦

28: **if** flag == false

29: $q_{proper} \leftarrow$ FindProperNode(T, q_{rand}, ρ)   ⎤

30: $U \leftarrow$ SolveCurve($T.g_{proper}$, q_{rand})   ⎪

31: **if** $U.r \geq r_{min}$ & CollisionFree(U, Q)   ⎬ Extend a tree

32: $T \leftarrow$ ExtendTree(T, U, q_{rand})   ⎦

33: $U \leftarrow$ SolveCurve($T.g_{rand}$, q_{goal})   ⎤

34: **if** $U.r \geq r_{min}$ & CollisionFree(U, Q)   ⎬ Get a path

35: $P \leftarrow$ GetPath(T, U, q_{goal})   ⎪

36: **end if**   ⎦

37: **end if**

38: **end while**

39: $p_{opt} \leftarrow$ Optimization(P)

40: **return** p_{opt}

GetPath(T, U, q_{goal})

1: $T \leftarrow$ ExtendTree(T, U, q_{goal})

2: $p \leftarrow$ ExtractPath(T)

3: P.add_path(p)

4: **return** P

ExtendTree(T, U, q)

1: $g \leftarrow$ GetConfig(U, q)

2: T. add_vertex(q)

3: T. add_edge(q, g)

4: **return** T


3.2 Optimization function

Among the candidate solutions, all of which have already met the required constraints, the optimal trajectory can be chosen based on a cost function. The objectives of the optimization take consideration of minimizing tissue trauma and the danger of the path, as well as the path shape evaluation.

$$\min F(\boldsymbol{u}, T) = \min\{\alpha_1 L + \alpha_2 D + \alpha_3 S\} \qquad (7)$$

where L is the length of path $L = \sum_{i=1}^{N} l_i$, D is the degree of danger in the path, $D = \min(D_i)$, where D_i is the distance between the path and the ith obstacle, S is the curve evaluation of the path and $S = \sum_{i=1}^{N} \frac{r_{min}}{r_i}$; α_1-α_3 are the weighted coefficients. The result of the function is the comprehensive evaluation of the optimal path.

4. Dynamic Motion Planning Algorithm

Prior to surgery, we first used the GHRG-RRTs in preoperative planning to seek out an optimal path based on
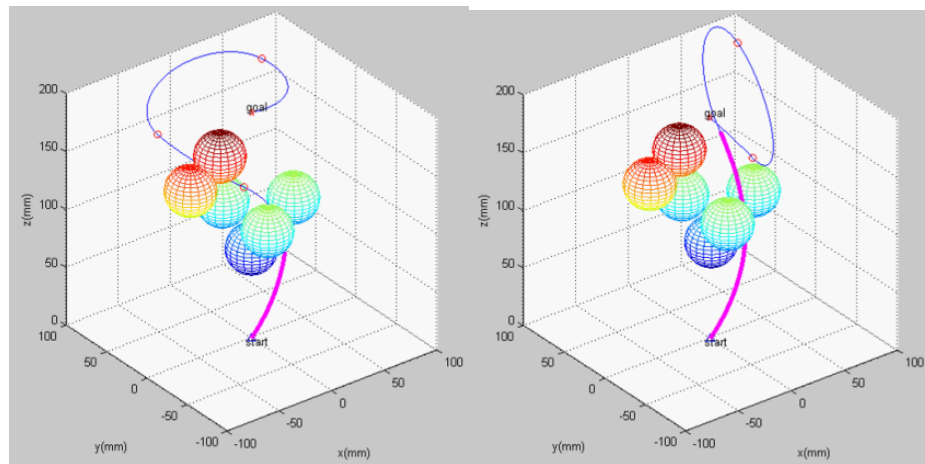
a) Large detour problem. b) Convergence problem.

Figure 6. Problems of the CMR

medical imaging (e.g., ultrasound, CT or MRI). During the surgery, we used online medical imaging to update the configurations of both the needle and the environment. We then used a fast intraoperative replanning algorithm to reform and adjust the path in order to realize a real-time closed-loop control up to the point where the needle reaches the target.

We acquired the optimal path using the preoperative motion planning algorithm. However, disturbances and uncertainties like model inaccuracy, tissue deformation and inhomogeneity, needle tip positioning errors, physiological movement, etc., could still deviate the needle from its intended path. Moreover, movement on behalf of the target and obstacles can render the planned path infeasible, leading to a collision with obstacles or misplacement of the target. To overcome this, we adopted the replanning idea [17], here referred to as conventional motion replanning (CMR), the outline of which is depicted in Algorithm 2. We attained the current state by using a 3D medical imaging system for replanning and execution in each cycle, in order to systematically rectify the path until the needle tip reaches target region δ.

Algorithm 2: CMR (g_{tip_now}, q_{goal_now}, Q_{now}).

1: $T \leftarrow$ InitTree(g_{tip_now}); $P \leftarrow P_{planned}$

2: **while** the distance between the needle tip and the goal $d > \delta$

3: $P \leftarrow$ Modified_GHRG (g_{tip_now}, q_{goal_now}, Q_{now})

4: **return** P

5: **end while**

---

where the modified GHRG-RRTs is the routine of lines 6-40 of Algorithm 1. Once the needle has been inserted into the tissue, the orientation at the needle tip can no longer be changed. Therefore, the linear path acquisition routine (lines 1-5 of Algorithm 1) are no longer appropriate for replanning.

This algorithm appears reasonable; however, it generally encounters some intractable problems. Firstly, this algorithm makes each cycle of replanning independent and without learning from former cycles, because of the stochastic property of the RRT algorithm, the blind replanning may cause a large detour in the path when the environment changes (see Fig.6a). Secondly, it may cause the convergence problem, i.e., as the flexibility of the needle is minimized due to nonholonomic constraints, whenever the needle gets close to the target, it is likely that the target will be unreachable because of its movement or other uncertain errors. Furthermore, the replanning algorithm will replan a circle-like path to make the target reachable again and as a result, the needle will detour in a loop without ever reaching the target (see Fig.6b). Therefore, the CMR algorithm is not suitable in terms of the significant movement of obstacles and the target.

To overcome these problems, we propose an improved motion replanning (IMR) algorithm with two strategies, i.e., an old point tracking strategy (OPTS) and an extreme trend extension strategy (ETES). OPTS tracks the old point in the old path in order to learn from former planning in order to overcome the large detour problem. ETES bends in an extreme manner when the needle gets close enough to a target that is unreachable to overcome the convergence problem. The programming of the IMR-1 is depicted in Algorithm 3.

OPTS (line 3-6 of IMR-1) tracks and utilizes the old connecting points q_i in the former path to replan a new path. If it cannot generate a new path using the old points, it will replan a new path by using a modified GHRG-RRTs. In this way, after several cycles of iteration, it will find a relatively stable path that is immune to changes in the environment to some extent, because it has learned from previous planning. This strategy overcomes the large detour problem.

```
Algorithm 3: IMR-1 (g_{tip_now}, q_{goal_now}, P_{planned}, Q_{now}).
─────────────────────────────────────────────────
1:  T ← InitTree(g_{tip_now}); P ← P_{planned}
2:  if P.length>s_1
3:    for all q_i∈P_{planned}
4:      U ← SolveCurve(T, q_i)
5:      T ← ExtendTree(T,U, q_i)          } Track the old points
6:    end for
7:    if all radii of T, r≥r_{min} & CollisionFree(T, Q_{now})
8:      P ← GetPath (T,q_{goal_now})
9:    else
10:     P ← Modified_GHRG (g_{tip_now}, q_{goal_now}, Q_{now})
11:   end if
12:   else if P.length≥Δs
13:     U ← SolveCurve(T,g_{tip_now}, q_{goal_now})
14:     if CollisionFree(U, Q_{now})
15:       if U.r<r_{min}
16:         U.r ← r_{min}
17:       end if
18:       q_{next} ← NextExtend (T,g_{tip_now},U.r,Δs)    } Extend in an extreme
19:       if ||q_{next}-q_{goal_now}||<||q_{tip_now}-q_{goal_now}||
20:         P ← GetPath(T,q_{extend})
21:       end if
22:     end if
23: end if
```

ETES (lines 14-21 of IMR-1) prevents the planner from generating a circle-like path due to the failure of reachability. If the needle gets close enough to a target that is unreachable (the radius of the last segment r is less than the minimum radius r_{min}), instead of replanning a new path, it will try its best to extend the needle to the target with the minimum radius until it gets to the closest position to the target. This strategy overcomes the convergence problem.

We performed different strategies according to different conditions. Here, s_1 is the switch between the proposed replanning strategies and $Δs$ is the length of the insertion we executed in each cycle. If the path was longer than s_1, the OPTS was performed; if the length of the path was between s_1 and $Δs$, the ETES was performed up to the point where the needle reaches the nearest position to the target.

Although the problems noted above were well solved, the ETES introduces another problem to the planning. Since we force the needle to extend to the target in an extreme manner, it may cause a large error (sometimes above 10mm), which is intolerant for clinical surgery. To overcome the complication, we propose the hardest goal tracking strategy (HGTS) and integrated it to the IMR-1 to form IMR-2, as depicted in Algorithm 4.

We found that the failure of reachability was badly influenced by movement of the target. Among all positions that the target has undergone, there is the hardest one to reach, which is contained in the path with the smallest radius. We called this the hardest goal. Theoretically, if the needle can achieve the hardest goal, it will be less difficult to achieve other positions linked to the goal. We therefore used the HGTS (lines 7-18 of IMR-2) to track the hardest

goal, regardless of the target movement, until the length of path was shorter than s_2.

```
Algorithm 4:IMR-2 (g_{tip_now}, q_{goal_now}, P_{planned}, Q_{now}).
─────────────────────────────────────────────────
1:  T ← InitTree(g_{tip_now}); P ← P_{planned}
2:  if P.length>s_2
3:    for all q_i∈P_{planned}
4:      U ← SolveCurve(T, q_i)
5:      T ← ExtendTree(T,U, q_i)          } Track the old points
6:    end for
7:    if all radii of T, r≥r_{min} & CollisionFree(T, Q_{now})
8:      P_{goal_now} ← GetPath(T,q_{goal_now})
9:      P_{goal_old} ← GetPath(T,q_{goal_old})
10:   else
11:     P_{goal_now} ← Modified_GHRG (g_{tip_now}, q_{goal_now}, Q_{now})
12:     P_{goal_old} ← Modified_GHRG (g_{tip_now}, q_{goal_old}, Q_{now})
13:   end if
14:   if the radius of the last segment of the two paths, respectively, r_{goal_now}< r_{goal_old}    } Track the hardest goal
15:     P ← P_{goal_now}
16:   else
17:     P ← P_{goal_old}
18:   end if
19:   else if P.length≥Δs
20:     U ← SolveCurve(T,g_{tip_now}, q_{goal_now})
21:     if CollisionFree(U, Q_{now})
22:       if U.r<r_{min}
23:         U.r←r_{min}
24:       end if
25:       q_{next} ← NextExtend (T,g_{tip_now},U.r,Δs)    } Extend in an extreme trend
26:       if ||q_{next}-q_{goal_now}|| < ||q_{tip_now}-q_{goal_now}||
27:         P← GetPath (T,q_{extend})
28:       end if
29:     end if
30: end if
```

5. Simulation and Discussion

5.1 Settings for simulation

We simulated the motion planner in MATLAB® (ver. 7.8.0, R2009a; MathWorks, Natick, MA) on a 2.5 GHz 4-core Intel® i5™ PC. We adopted an environment similar to [14], modelling it as a cubical region at 200mm along each axis. Six spherical obstacles, each with a radius of 20mm represent the pubic arch, the urethra and the penile bulb around the prostate. The goal is set to (0 0 195), in millimetre (see Fig.8a). We set the minimum radius r_{min}=50mm and the specific metric at $ρ$=10mm (in line 29 of Algorithm 1). The maximum number of the candidate paths was set to 100 and the maximum number of iterations to 10 000. In order to speed up computation, we assumed there was a relatively safe margin m (here we set m=3mm) around the obstacle; as long as the needle did not puncture the safe margin, it would never puncture the obstacle; thus, the surgery was safe. The radius of the obstacle should also be enlarged by safe margin m when planning to make it safe. As such, there was no need to care about the exact distance between the needle and the obstacles, as long as the surgery was sufficiently safe. Therefore, we could disregard the second term in (7) by setting $α_2$=0. Other weighted coefficients were set to $α_1$=$α_3$=1, as we equally considered both of the remaining sub-objective functions. These settings were

Algorithms	Large detour problem (times)	Convergence problem (times)	Error (mm)	Time for each cycle (ms)	Length of optimal path (mm)	Length of actual path (mm)
CMR	18	16	/	202.6±101.1		/
IMR-1	0	0	7.77±4.15	1.6±11.4	214.5±5.6	228.6±18.2
IMR-2	0	0	1.31±0.38	3.7±21.6		219.8±4.7

Table 1. Comparison of the three algorithms

applied to both preoperative planning and intraoperative replanning.

For replanning, we added some disturbances and movements. The disturbances such as model inaccuracy, tissue deformation, tissue inhomogeneity and needle tip positioning errors, were modelled as white noise and followed normal distribution N~(μ, σ). We let μ=r and σ=r/10 mm to the radii of the path, μ=0 and σ=1mm to the tip position and μ=0 and σ=0.01 rad to the orientation of the needle tip. Movement of the target and obstacles were modelled as periodic sinusoidal motions in 3D, both with an amplitude of 5mm and at a period of 60s and 5s, respectively.

As noted, since s_i(i=1,2) are the switch for the replanning strategies and affect the validity of the planning, we need to discuss them in detail. For IMR-1, s_1 is the switch between OPTS and ETES. Here we will track the old connecting points along the planned path. The convergence problem always occurs in the final segment, especially close to the target. We thus have to place s_1 before the convergence problem occurs in the final segment. In order to attain switch s_1, we have to study where the convergence problem generally occurs and how long the final segment of path is. We simulated for 20 times of the Algorithm 2 (CMR) in the specified environment. Among the 20 trials, the maximum length of path occurring in the convergence problem was 30.73 mm; the minimum length of the final segment was 92.01 mm. Thus, s_1 should be between 30.73 mm and 92.01 mm; to be safe, we let s_1= 40~80 mm. Moreover, the convergence problem is also related to the environment and the value of minimum radius r_{min}.

As for IMR-2, s_2 is the switch between the first two strategies (OPTS and HGTS) and the last one (ETES). The analysis for IMR-2 is similar to that of IMR-1. We will perform the first two strategies until the needle gets close to the target and then perform the final strategy. However, the difference from the IMR-1 is that we do not have to worry about the convergence problem prior to releasing the HGTS, since the hardest goal is fixed. The critical matter is when to release the HGTS. If we release it too early, it will track the target thereafter and the effectiveness will be no better than for IMR-1; if we release it too late, the planning will have limited opportunity to adjust the path to the current target. Thus, the value of s_2 for IMR-2 is neither too large nor too small. In order to attain switch s_2 for IMR-2, we simulated each different value 20 times to observe its performance, as is shown in Fig.7. From the figure, we can conclude that the best performance for IMR-2 is at s_2=10. Moreover, s_2 not

only concerns the environment and r_{min}, but also target movement.

Therefore, empirically, we set s_1=50 mm and s_2=10 mm for IMR-1 and IMR-2, respectively. We also set the insertion distance per cycle as Δs = 1 mm.
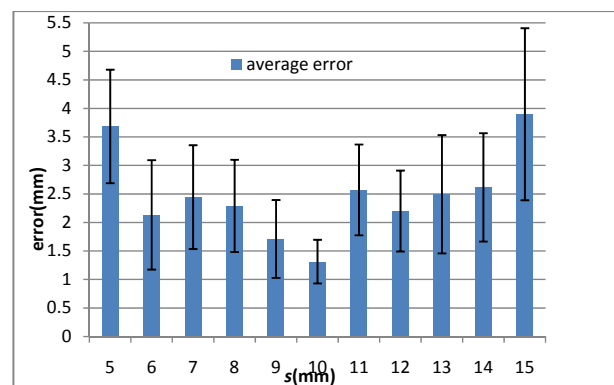

Figure 7. Average errors of different values of s_2

5.2 Simulation results and discussion

In order to verify the validity of each strategy, we compared Algorithm 2 (CMR), Algorithm 3 (IMR-1) and Algorithm 4 (IMR-2). We performed 20 trials using the same target for each algorithm. The validity of the OPTS and ETES is obvious in the comparison of CMR and IMR-1, as shown in Table 1. The validity of the HGTS is shown in the comparison of IMR-1 and IMR-2 (see Table 1). Table 1 shows that some of the results in the 20 trials have the formation "mean ± standard deviation".

The results show that CMR suffers from the large detour problem, while the other algorithms avoid it. This reveals that OPTS is effective in terms of the large detour problem. CMR also suffers the convergence problem, while the other algorithms avoid this, which reveals the ETES is effective in terms of the convergence problem. Moreover, the time for each cycle for CMR is much longer than that for the other algorithms. This is because OPTS, which not only learns from former cycles of replanning and renders the path stable, but also enhances the cycling replanning speed.

On the other hand, from the comparison of IMR-1 and IMR-2, it is clear that the accuracy of IMR-2 is much better than that of IMR-1, which reveals the effectiveness of the HGTS. The error of IMR-2 is within 2mm, which meets the needs of clinical requirements. From the length of the actual path, we can see that IMR-2 is much more stable than

IMR-2	Large detour problem (times)	Convergence problem (times)	Error (mm)	Time for each cycle (ms)	Length of optimal path (mm)	Length of actual path (mm)
G1	0	0	1.56±0.53	2.5±18.3	204.9±6.1	208.2±8.7
G2	0	0	1.47±0.37	3.3±22.1	197.1±5.2	199.0±4.2

Table 2. The performance of the IMR-2 algorithm

a) The original optimal path.

b) Insertion at 20mm.

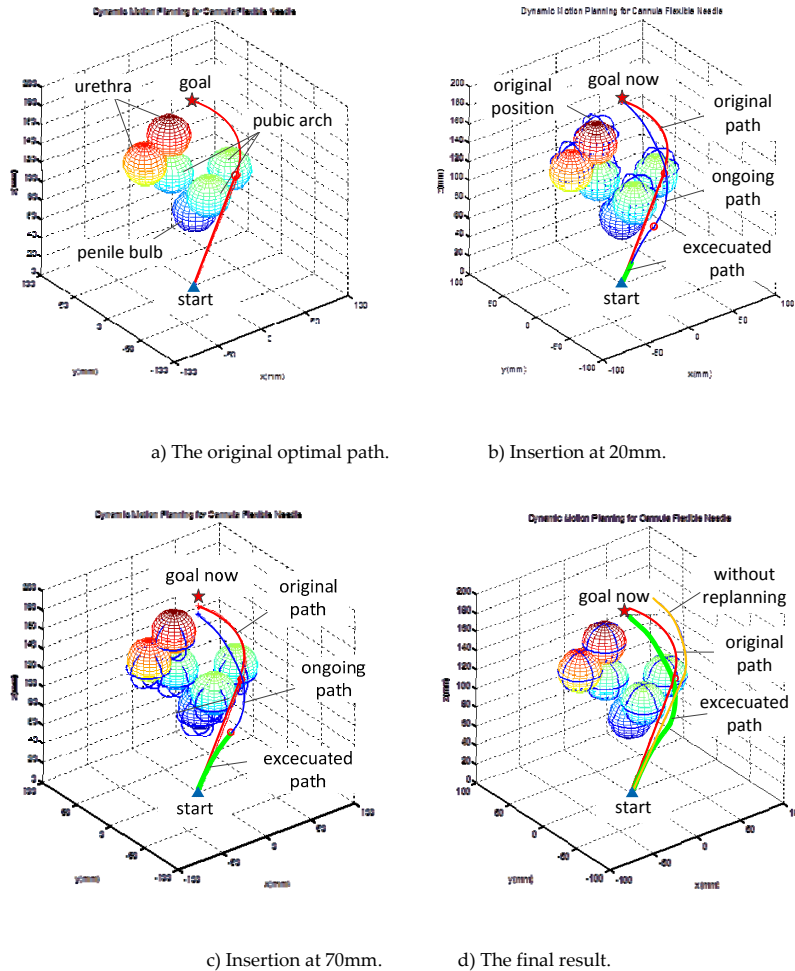c) Insertion at 70mm.

d) The final result.

Figure 8. Simulation result of Algorithm 4

IMR-1. Although the replanning speed of IMR-2 is slightly slower than that of IMR-1 due to adding the extra strategy of HGTS, it is still in milliseconds (see the column "Time for each cycle" in Table 1), which is fast enough for intraoperative replanning.

We also obtained one of the simulation results and compared it with the result of the simulation without the replanning. The course of simulation is shown in Fig.8. The original optimal path, the executed path, the ongoing/adjusted path and the path added with disturbances and without replanning or adjusting are depicted in the figure. The final errors were 0.75 mm and 43.15 mm for the replanning and without replanning paths, respectively.

In order to verify the robustness of the proposed algorithm (IMR-2), we also performed simulation for different goals. We randomly set two more goals as G1 (5 20 190) and G2

(10 -20 185), both in millimetre, and simulated each goal 20 times. The results are as shown in Table 2 and have the formation "mean ± standard deviation".

From Table 2, we can see that the large detour or convergence problem did not occur, the error is within 2mm and speed remains high. As such, the conclusion can be drawn that the proposed algorithm has strong robustness.

6. Conclusion

In this work, we firstly calculated the kinematics for the cannula flexible needle. Based on the proposed RGHG-RRTs algorithm for static motion planning in our previous work, we propose a 3D dynamic motion planning algorithm for intraoperative surgery. We also propose three novel strategies to be integrated in the conventional

replanning algorithm in order to improve it. By comparing the CMR, the IMR-1 and the IMR-2 algorithms, we can conclude that the proposed OPTS, ETES and HGTS are extremely effective for addressing the large detour problem, the convergence problem and accuracy problem, respectively. Moreover, the OPTS also benefits for the replanning speed of a cycle. Finally, we performed simulations for different goals. The results revealed the validity and robustness of the proposed replanning algorithm.

In future, we will integrate our planning with a real-time feedback controller to carry out experiments on the phantom tissue.

7. Acknowledgements

8. References

[1] Webster III RJ, Kim JS, Cowan NJ, Chirikjian GS, Okamura AM. Nonholonomic modeling of needle steering. Int. J. Robotics Research. 2006; 25(5-6), pp. 509-525.

[2] Minhas DS, Engh JA, Fenske MM. Modeling of needle steering via duty-cycled spinning. Proc. Int. Conf. IEEE Eng. in Medicine and Biology Society; 2007; pp. 2756-2759.

[3] Majewicz A, Siegel JJ, Stanley AA, Okamura AM. Design and Evaluation of Duty-Cycling Steering Algorithms for Robotically-Driven Steerable Needles. IEEE International Conference on Robotics & Automation; 2014; pp. 5883-5888.

[4] Datla NV, Honarvar M, Nguyen AM, KonhB, Darvish K, Yu Y, et al. Toward a nitinol actuator for an active surgical needle," in ASME Conference on Smart Material, Adaptive Structures and Intelligent Systems, 2012; pp. 19-21.

[5] Konh B, Honarvar M and Hutapea P. Application of SMA wire for an active steerable cannula. ASME Conference on Smart Materials, Adaptive Structures and Intelligent Systems, 2013; pp. 265-269.

[6] Honarvar M, Datla NV, Konh B, Podder T K, Dicker AP, Yu Y, Hutapea P. Study of unrecovered strain and critical Stresses in one-way shape memory nitinol. Journal of Materials Engineering and Performance, 2014; 23(8), pp. 2885-2893.

[7] Duindam V, Alterovitz R, Sastry S, Goldberg K. Screw-based motion planning for bevel-tip flexible needles in 3D environments with obstacles. Pro-

ceedings of the IEEE International Conference on Robotics and Automation; 2008; pp. 2483-2488.

[8] Alterovitz R, Branicky M, Goldberg K. Motion Planning under Uncertainty for Image-guided Medical Needle Steering. The International Journal of Robotics Research, 2008; 27(11-12), pp. 1361-1374.

[9] Park W, Wang Y, Chirikjian GS. The path-of-probability algorithm for steering and feedback control of flexible needles. International Journal of Robotics Research; 2010; 29(7), pp. 813-830.

[10] Alterovitz R, Goldberg K, Okamura A. Planning for steerable bevel-tip needle insertion through 2D soft tissue with obstacles. Proceedings of the IEEE International Conference on Robotics and Automation; 2005; pp. 1640-1645.

[11] Patil S, Berg JVD, Alterovitz R. Motion planning under uncertainty in highly deformable environments. Proc. Robotics: Science and Systems (RSS), June; 2011.

[12] Alterovitz R, Patil S, Derbakova A. Rapidly-exploring Roadmaps: weighing exploration vs. refinement in optimal motion planning. In: Proc. IEEE Int. Conf. on Robotics and Automation; 9-13 May 2011; Shanghai, China. 2011; pp. 3706-3716.

[13] Lobaton E, Zhang J, Patil S, Alterovitz R. Planning curvature-constrained paths to multiple goals using circle sampling. In: Proc. IEEE Int. Conf. on Robotics and Automation; 9-13 May 2011; Shanghai, China. 2011; pp. 1463-1469.

[14] Xu J, Duindam V, Goldberg K. Motion planning for steerable needles in 3D environments with obstacles using rapidly exploring random trees and backchaining. In: Proc. IEEE Int. Conf. on Automation Science and Engineering; 2008 Aug 23-26; Arlington, VA, USA. 2008; pp. 41-46.

[15] Patil S, Alterovitz R. Interactive motion planning for steerable needles in 3D environments with obstacles. In: Proc. IEEE RAS/EMBS Int. Conf. Biomedical Robotics and Biomechatronics; 2010 Sep 26-29; Tokyo, Japan. 2010; pp. 893-899 (2010).

[16] Caborni C, Ko SY, Momi ED, Ferrigno G, Baena FRY. Risk-based path planning for a steerable flexible probe for neurosurgical intervention. In: IEEE RAS/EMBS Int. Conf. on Biomedical Robotics and Biomechatronics; 24-27 Jun 2012; Rome, Italy. 866-871 (2012).

[17] Patil S, Burgner J, Webster III RJ, Alterovitz R. Needle steering in 3-D via rapid replanning. IEEE Trans. on Robotics. 2014; 30(4), pp. 853-864.

[18] Vrooijink GJ, Abayazid M, Patil S, Alterovitz R, Misra S. Needle path planning and steering in a three-dimensional non-static environment using two-dimensional ultrasound images. Int. Journal of Robotics Research. 2014; 33(10), pp. 1361-1374.

[19] Bernardes MC, Adorno BV, Poignet P, Borges GA. Robot-assisted automatic insertion of steerable

needles with closed-loop imaging feedback and intraoperative trajectory replanning. Mechatronics. 2013; 23(6), pp. 630-645.

[20] Bernardes MC, Adorno BV, Borges GA, Poignet P. 3D robust online motion planning for steerable needles in dynamic workspaces using duty-cycled rotation. Journal of Control, Automation and Electrical Systems. 2014; 25(2), pp. 216-227.

[21] Zhao YJ, Joseph FOM, Yan K, et al. Path Planning for Robot-Assisted Active Flexible Needle using Improved Rapidly-Exploring Random Trees. 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'14), Chicago, Illinois, USA, August 26-30, 2014; pp. 380-383.

[22] Zhao YJ, Konh B, Honarvar M, et al. 3D Motion Planning for Robot-Assisted Active Flexible Needle Based on Rapidly-Exploring Random Trees. Journal of Automation and Control Engineering. 2015; 3(5), pp. 360-367.