Planning Curvature-Constrained Paths to Multiple Goals Using Circle Sampling

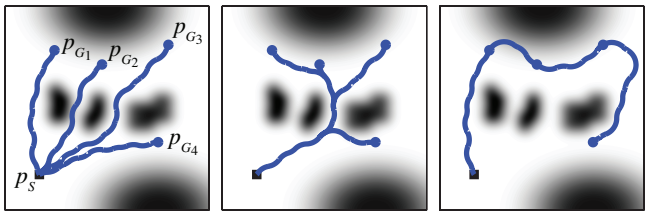
Edgar Lobaton, Jinghe Zhang, Sachin Patil, and Ron Alterovitz

***Abstract*— We present a new sampling-based method for planning optimal, collision-free, curvature-constrained paths for nonholonomic robots to visit multiple goals in any order. Rather than sampling configurations as in standard sampling-based planners, we construct a roadmap by sampling circles of constant curvature and then generating feasible transitions between the sampled circles. We provide a closed-form formula for connecting the sampled circles in 2D and generalize the approach to 3D workspaces. We then formulate the multi-goal planning problem as finding a minimum directed Steiner tree over the roadmap. Since optimally solving the multi-goal planning problem requires exponential time, we propose greedy heuristics to efficiently compute a path that visits multiple goals. We apply the planner in the context of medical needle steering where the needle tip must reach multiple goals in soft tissue, a common requirement for clinical procedures such as biopsies, drug delivery, and brachytherapy cancer treatment. We demonstrate that our multi-goal planner significantly decreases tissue that must be cut when compared to sequential execution of single-goal plans.**

I. Introduction

We consider a variant of nonholonomic motion planning in which a robot, with motion subject to curvature constraints, must visit a set of goals. The robot can visit these goals in any order but must visit each goal at least once. This problem arises in a variety of applications, such as a wheeled personal assistant robot that must pick up multiple objects from the floor, a mobile robot that must inspect a set of sites, and a medical robot that must perform a biopsy at multiple locations within a suspected tumor. It is also often important to consider the optimality of computed motion plans in terms of application-specific metrics such as minimizing distance traveled by the robot, time taken to complete the task, or tissue damage in a medical procedure.

The optimal multi-goal plan depends on the properties of the nonholonomic robot and the objective function that defines costs to be minimized. We consider distinct costs for forward motion and for retracing a previously traversed segment. If retracing a previously traversed segment is not permitted (i.e. has infinite cost), then the optimal solution takes the form of a curve in space. If retracing a path is permitted, then the optimal solution takes the form of a tree where the nonholonomic robot can move forward and backward along segments of the tree. The algorithms we propose uses a general cost metric that encompasses both of these cases. We also allow the use of cost maps to associate cost with every spatial location. As shown in Fig. 1, the plans produced by our multi-goal planner result in substantially less cost than sequentially executing independent plans from the start location to each goal.



(a) Independent paths (b) Optimal path, retrace at no cost (c) Optimal path, retrace disallowed

Fig. 1. Multi-goal plans from point p_S to goals p_{G_k} computed using our circle sampling approach. The objective is to minimize the length of the path and costs associated with closeness to obstacles (grayscale values) while assuming that retracing previously traversed locations does not incur any cost (as is the case with steerable needles). (a) Plan found by computing paths from p_S to each goal independently. (b) Optimal plan found by solving a directed Steiner tree problem (32% better than independent paths). (c) Plan found by disallowing path retrace (18% better than independent paths).

We focus on the application to medical steerable needles, a recently developed class of bevel-tip flexible needles that can be steered along curved paths to reach previously inaccessible goal locations [25], [1]. The need to reach multiple goal locations in soft tissue is a common requirement for clinical needle-based procedures such as biopsies, drug deliveries and brachytherapy cancer treatment. A naïve solution would be to sequentially execute many ab initio single goal plans, which causes unnecessary trauma to the tissue. This can be avoided by deploying a steerable needle to one of the goals, partially retracting to a specified location within tissue, and then steering the needle along a different path to reach a new goal. This can be repeated until all the clinical goal locations are reached. This model of needle insertion is equivalent to the case in Fig. 1. It is reasonable to assume from a clinical perspective that tissue is only cut when the needle is inserted and not while it is being retracted. The planning objective in this case is to limit tissue damage by minimizing the total extent of tissue cut by the needle-tip (i.e. minimizing forward movement distance) during the procedure.

Computing an optimal curvature-constrained path that visits multiple goals is challenging because it combines the problem of nonholonomic motion planning with the problem of determining the order in which the goals should be visited. Solving these problems to optimality with completeness guarantees requires computation time that grows exponentially with the complexity of the environment and the number of goals. The general problem of nonholonomic motion planning with obstacles is known to be PSPACE hard

All authors are with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA {lobaton,jing2009,sachin,ron}@cs.unc.edu

[13], while the goal ordering problem is equivalent to the directed Steiner tree problem which is NP-complete [3].

We make two key contributions. *First*, we introduce a new subpath-based sampling approach to compute curvature-constrained paths that generalizes to both 2D and 3D environments. Instead of sampling points in the configuration space and attempting to connect them, our method constructs a roadmap by randomly sampling circles of bounded curvature and generating feasible transitions between these circles. The resulting roadmap is used to solve multi-goal queries by computing a directed Steiner tree over the roadmap, which is a minimum-cost tree rooted at the start configuration that visits all the goals. *Second*, we introduce greedy heuristics that reduce the exponential complexity of multi-goal planning and quickly compute approximate plans that work well in practice. The computational complexity of the heuristics is $O(N_G\,|E| + N_G\,|V|\,\log|V|)$, where N_G is the number of goals and V and E are the set of vertices and edges in the roadmap, respectively.

We apply our method to compute curvature-constrained paths in environments with obstacles. We present results in 2D as well as preliminary results in 3D. We demonstrate that the proposed greedy heuristics converge to within 5% of the optimal solution and offer significant improvements as compared to the naïve approach of sequentially executing multiple single-goal plans.

II. Related Work

Nonholonomic motion planning is a well-studied area in robotics and related fields [4], [13]. Prior work has analyzed optimal paths for curvature-constrained mobile robots using a discrete set of canonical trajectories [6], [22], [9]. These approaches have been extended to compute optimal paths in workspaces with obstacles using graph search algorithms on grid-based cost maps [15], [20]. These methods do not scale well to higher dimensions and only ensure optimality up to the chosen grid resolution. Other approaches compute optimal paths using variational methods to solve a two-point boundary value problem [12], [2]. Variational methods are computationally expensive and may suffer from numerical issues.

Sampling-based motion planning has become increasingly popular in recent years [13]. Extensions to the probabilistic roadmap planner (PRM) [11] and rapidly-exploring random trees (RRT) algorithm [14] have been proposed to consider nonholonomic robots with differential constraints. Instead of sampling points in the configuration space and connecting them like in PRMs or RRTs, our method samples feasible subpaths in the configuration space by sampling circles of bounded curvature and generating feasible transitions between these circles. Our method is applicable to a wide variety of robots with steering mechanisms and generalizes to both 2D and 3D workspaces.

We focus on the application of steering flexible, bevel-tip needles through tissue for clinical procedures such as biopsies and drug injections. These needles naturally follow paths of constant curvature [25], which can also be varied using duty-cycling [16]. Single-query motion planning for steerable needles has been extensively studied for 2D workspaces [1] and 3D workspaces [10], [7], [18], [19]. These methods do not address optimal, multi-goal planning.

Often, a steerable needle has to reach multiple goals within a region of interest such as a tumor. Multi-goal motion planning also arises in other domains such as wheeled personal assistant robots picking up multiple objects off the floor and industrial robots performing coordinate measurements [24], site inspections [5], drilling [8] and spot-welding [23]. Once an appropriate roadmap is obtained, the multi-goal planning problem is equivalent to the minimum directed Steiner tree problem. Saha et al. [23] use an approximate solution provided by Chekuri et al. [3] to solve the multi-goal planning problem for a holonomic robot arm used for spot-welding. Recently, Elinas [8] used genetic algorithms (GA) in combination with a PRM planner to compute an approximate Hamiltonian tour for a car-like nonholonomic blasthole drill for mining operations. In contrast, our approach allows the robot to retrace configurations in the roadmap, as is the case with needle steering applications. Our approach is easier to implement and computationally efficient as compared to the variational solver [12] used by Elinas [8] to construct edges in the roadmap.

III. Problem Definition

We consider a nonholonomic robot modeled as a simple car with a minimum turning radius r (equivalent to maximum curvature $\kappa = 1/r$). This model is applicable to bevel-tip steerable needles being controlled in an imaging plane [1]. We model the environment as a 2D domain with polygonal obstacles. The robot's configuration q is defined by its position (x, y) and its orientation θ on the plane. The robot is constrained to begin at a position p_S at any orientation. As the robot moves, it traces a path $\gamma(t) : [0, T] \rightarrow \mathbb{R}^2$ parameterized by arc-length $t \in [0, T]$. The curvature of the path is bounded in magnitude by constant κ. As the robot moves along $\gamma(t)$, it accumulates costs $f(\gamma(t))$, where the positive cost function $f : \mathbb{R}^2 \rightarrow \mathbb{R} \cup \{\infty\}$ is defined over the 2D environment. The cost metric is application specific and can represent quantities such as distance traveled, clearance from workspace obstacles, time taken to complete the task, or extent of tissue damage. If the robot retraces a point $\gamma(t)$ in the domain, then it incurs cost $g : \mathbb{R}^2 \rightarrow \mathbb{R} \cup \{\infty\}$ in subsequent instances, which may or may not be equal to f depending on the application. The total cost J for a path γ is hence:

$$J(\gamma) = \int_0^T \left\{ \begin{array}{ll} f(\gamma(t)) & \text{if } \forall t' < t, \gamma(t) \neq \gamma(t') \\ g(\gamma(t)) & \text{otherwise} \end{array} \right\} dt \tag{1}$$

The path begins at p_S, so $\gamma(0) = p_S$. Depending on the application, we may require that $\gamma(T)$ terminate at p_S. The objective is to compute a path γ that visits a set of N_G goal points $p_{G_1}, p_{G_2}, \ldots, p_{G_{N_G}}$ and minimizes J.

We note that the flexibility in defining f and g on a problem-specific basis allows us to consider a wide variety of applications. If our objective is to find the shortest path

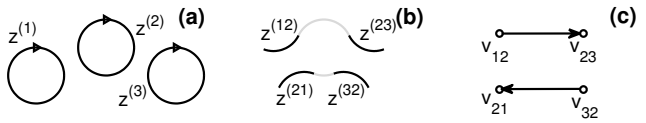


Fig. 2. Constructing a roadmap. (a) A set of sampled subpaths. (b) Bridges and subpath segments joining them to form a roadmap. (c) Graphical representation of the roadmap.

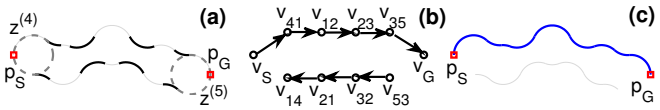


Fig. 3. Finding the shortest path between points p_S and p_G by using the roadmap from Fig. 2. (a) Circles passing through the start and goal locations are added, and arc bridges connecting these new circles are found. (b) The graph model with the new vertices and edges. (c) The shortest path in the graph corresponds to a path in the physical space that satisfies the curvature constraints.

while avoiding obstacles, then we define the cost functions f and g to have value 1 in the feasible space of the environment and ∞ at regions occupied by the obstacles. Setting $g = 0$ corresponds to ignoring the cost of retracing a path. This setting is particularly relevant to quantifying tissue damage during needle steering since retraction of the needle does not damage new tissue. This type of problem is illustrated in Fig. 1 (b). Another variant of this problem corresponds to setting $g = \infty$, which prohibits retracing previously traversed subpaths. An example of such a plan is illustrated in Fig. 1 (c).

In keeping with the application of needle steering, for simplicity of presentation we will assume throughout the rest of the paper that forward motion incurs cost ($f(x) = 1$) while retracing a path incurs no cost ($g(x) = 0$). However, our formulation can be easily modified to account for any of the previously mentioned conditions. Under our assumptions, a needle can be deployed to one of the goals, retracted to a specified location, and then deployed again along a different path to reach a new goal. The process can be repeated until all goals are reached. The total cost of the plan is the total length that the needle was deployed (without counting retractions).

In section IV, we consider the special case of $N_G = 1$. In section V, we generalize to any N_G. We present results in a 2D environment in section VI. We then extend our formulation to the case of 3D environments and present results in 3D in section VII.

IV. Single Goal Planning

A. General Framework

Sampling-based methods for path planning traditionally take the approach of sampling the robot's configuration space and finding control sequences or local plans that connect the sampled configurations. Instead of sampling points in the configuration space, our approach is to sample feasible subpaths and connect them.

We begin by sampling *subpaths* that satisfy the nonholonomic constraints on the robot's motion. Due to the curvature constraint in our problem, we choose to sample circles of radius $1/\kappa$. Paths constructed with segments of constant curvature can be used to approximate arbitrary paths with bounded curvature. We then create a roadmap using a local planner that connects subpaths together. Given subpaths $z^{(i)}$ and $z^{(j)}$, we denote the *bridge* between i and j, $z^{(ij)}$, as their connecting path. We note that our selection of circles as subpaths is one of many options. Selecting subpaths with 0 time duration would correspond to single points in the configuration space and the constructed roadmap becomes a traditional PRM. We show that by selecting full circles as our subpaths, it is easier to find connections between the subpaths and that a small number of subpaths will be needed to yield a highly connected roadmap.

In this section, we discuss how to construct a roadmap and demonstrate how to find an optimal path γ connecting a start point p_S to a goal point p_G.

B. Constructing a Roadmap using Sampled Circles

We begin by sampling the centers of circles with radii $1/\kappa$. Any random or pseudo-random sampling could be used; our implementation uses a Halton sequence [13] in order to obtain a disperse and uniform sampling of the 2D domain. We assume that the robot's motion through a circle C is in the clockwise direction, as shown in Fig. 2 (a).

After sampling the circles, we must create bridges between the circles. Given circles C_i and C_j we can find a circle C_{ij} tangential to them as long as the distance between their centers is less than $4/\kappa$. Note that in most cases there exist two such connecting circles and we always choose the one with the shortest length as the bridge $z^{(ij)}$. The subpaths are now segments of the original sampled circles, and the bridges are segments of the connecting circles, as shown in Fig. 2 (b).

It is clear that given this construction, the paths in the roadmap will satisfy the model constraints from Section III. Note that the bridges define counter-clockwise paths that are only capable of connecting clockwise subpaths. If we consider counter-clockwise subpaths, these would not be able to connect to their clockwise counterparts, which would generate two disjoint roadmaps. Hence, it is important that we only consider one consistent orientation for the subpaths. The process is illustrated in Fig. 2 (b) in which bridges are marked as dark circular segments and the segments of the subpaths connecting them are marked as gray segments.

A graphical representation of the roadmap is obtained by representing each bridge $z^{(ij)}$ as a vertex and subpath segments joining them as directed edges as shown in Fig. 2 (c). We note that bridges and subpaths that intersect obstacles are omitted from the graph.

C. Adding Start and Goal Points

Next, we solve for a path connecting a start point p_S to a goal point p_G. The basic approach is to add vertices v_S and v_G to the graphical representation and connect to bridges whenever possible. We also consider the case where the orientation of the path at p_S and p_G are variables that can

be optimized. In this case, we sample circles passing through p_S and p_G at different allowed orientations. Then, we add the vertices v_S and v_G, and any resulting bridge from the newly sampled circles. Alternatively, we can enforce a given orientation from the entry point by sampling a single circle at the appropriate position.

This process is illustrated in Fig. 3 in which circles $z^{(4)}$ and $z^{(5)}$ are added, and a new roadmap and graph representation are computed.

D. Computing an Optimal Solution

Our objective is to compute a path that minimizes the cost function J. We can transform this optimal path planning problem into a graph problem using the roadmap constructed above. The weights for the vertices and edges in our graphical representation would correspond to the function f or g integrated over the arc-length of the associated circular segment. The vertices v_S and v_G receive a weight of 0. Once this graph has been constructed, we can apply standard graph shortest path algorithms to find the path with the lowest cost. A simple example is illustrated in Fig. 3 (c).

As the number of sampled circles increases, the probability that this method returns a feasible solution when one exists approaches 1 assuming that there exists a path with ϵ clearance away from the obstacles. We can argue this probabilistic completeness in the following manner. Let γ' be a path connecting two locations p_S and p_G in the environment which satisfies our kinematic constraints. Path γ' must be formed by segments of circles at given locations with alternating orientations of the arc. Let $\{o_i\}$ be the collection of the centers of the circles with clockwise orientation. Perturbing the path by small amounts (which assumes that we have at least an ϵ clearance around the path) will slide the points o_i creating small neighborhoods around them. We can form a path joining the points p_S and p_G by sampling circles from these neighborhoods, which will occur with increasing probability as the number of samples increases due to our uniform sampling of the space.

V. Multi-Goal Planning

In this section we extend the result of the previous section to search for paths starting at location p_S and reaching multiple goal points $\{p_{G_k}\}_{k=1}^{N_G}$, where N_G is the number of goals.

The formulation of the problem for multiple goals is similar to the single-goal case. A weighted graph can be constructed in the same way as before. Our objective in this case is to find a path rooted at v_S such that the path visits all of the goals and minimizes the sum of the costs of the arcs visited.

The solution to this problem is the same as the solution for the Steiner directed tree problem, which is defined as finding the minimum cost directed tree rooted at v_S and spanning all of the goal vertices. Unfortunately, it is known that the Steiner tree problem is NP-complete and several approximation techniques are available in the literature [3].

Algorithm 1 Complete Algorithm

Require: A distance matrix D, a root vertex v_S, a set of goal vertices $V = \{v_1, v_2, \cdots v_N\}$.

1: Set the list of edges forming the tree $T = \emptyset$, and set the length of the tree $d = \infty$.
2: **if** $|V| = 1$ **then** add (v_S, v_1) to T, set $d = D(v_S, v_1)$ and go to Step 11.
3: **for** all partitions of V into sets V_a and V_b **do**
4: **for** all vertices v_0 in the graph **do**
5: Get T_a and d_a from a recursive call to the algorithm using D, v_0 and V_a as inputs.
6: Get T_b and d_b from a recursive call to the algorithm using D, v_0 and V_b as inputs.
7: Let $d' = d_a + d_b + D(v_S, v_0)$.
8: **if** $d' < d$ **then** set $d = d'$ and set T to be $\{(v_S, v_0)\} \cup T_a \cup T_b$.
9: **end for**
10: **end for**
11: **return** T and d.

Algorithm 2 Shortest 1st Heuristic Algorithm

Require: A weighted graph W, a root vertex v_S, a set of goal vertices $V = \{v_1, v_2, \cdots v_N\}$.

1: Set the list of edges forming the tree $T = \emptyset$, and set the length of the tree $d = 0$.
2: **while** $|V| > 0$ **do**
3: Choose the goal in V with the smallest cost from v_S using W. Let it be v_0 with path P.
4: Add the cost of P to d.
5: Set $T = T \cup P$ and remove v_0 from V.
6: Set the weight in W of every edge in P to 0.
7: **end while**
8: **return** T and d.

A complete solution to this problem can be obtained by exploiting the fact that the solution has to be a tree. We note that the solution must start from vertex v_S as a single path and then split into two paths (we account for a single path splitting into n paths at a vertex by having $n-1$ two-way splits at the same vertex). Each split path will reach a subset of the goals. These subsets form a partition. Each path then splits again, and the process is repeated until there is a single goal per path. The process is illustrated in Algorithm 1 in which we assume that D is a matrix storing the distance between any pair of points in the graph. The tree is specified by a collection of edges in D, where each edge represents the shortest path between the corresponding vertices.

Given that there is a total of N_G goals and N_v vertices in the graph, there are $2^{N_G-1}-1$ possible partitions of the goals which yields a total of $N_v(2^{N_G-1}-1)$ recursive iterations in the first level. Since for every new level k of recursion the algorithm will need less than $N_v(2^{N_G-k}-1)$ iterations, an upper bound to the number of iterations for the algorithm is $(N_v(2^{N_G-1}-1))^{N_G-1}$. Note that the complexity increases dramatically with the number of goals.

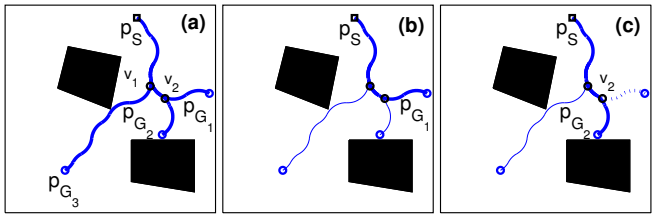


Fig. 4. Plan execution for a steerable needle. (a) Goals are numbered using a depth first approach on the tree selected from the graph. (b) The goal p_{G_1} is visited. (c) The needle is retracted to junction point v_2 and then deployed towards p_{G_2}. The process can then be repeated to get to p_{G_3}.

Due to the complexity of searching for the optimal solution to our problem, we will consider several heuristics. In Algorithm 2, we introduce the **Shortest 1st** heuristic that selects paths to the goals by choosing the paths with the smallest distance first. We also define the **Longest 1st** heuristic that selects the longest distance path first. Note that both of these approaches only require N_G computations of the shortest path in the graph, which yields a computational complexity of $O(N_G\,|E|+N_G\,|V|\,\log|V|)$. We also introduce the **Combinatorial** heuristic in which we try all possible orderings (instead of selecting the shortest or longest) and select the ordering with the shortest length. Note that this approach requires $N_G\,!$ times more computations.

Since our solutions will be a directed tree rooted at v_S, it is straightforward to determine how to execute a plan for visiting the goal vertices. All that is required is to label all vertices in the directed tree using a depth-first approach and visit the goals in the order in which they appear by first moving the robot to the first goal and then retracting until the robot gets to the vertex from which it can branch off to the next goal. The process is illustrated in Fig. 4. Using a depth-search approach the goals are labeled from p_{G_1} to p_{G_3}. After visiting the first goal, the robot retracts to the position of vertex v_2 and then moves to goal p_{G_2}. The process is repeated one more time by retracting to v_1 and then moving to goal p_{G_3}.

VI. Results

We analyze the performance of our approach to multi-goal planning using the environments shown in Fig. 1 and Fig. 4. We model the environment as a unit square. The minimum permissible radius of curvature is set to 0.1 units. We add 4 tangential circles at the start point and each goal (as described in Section IV).

In order to experimentally analyze the probabilistic completeness of our algorithm, we study the effect of the density of the subpaths on the connectivity of the roadmap for the environment in Fig. 4. To experimentally consider the effect of different situations, we independently select points in each experiment for the start and goal locations using a uniform random distribution. Fig. 5 (left) illustrates the proportion of times in which the algorithm was able to find a path between arbitrary entry points and goals. Each point in the plot is computed using 500 iterations of random sampling. As observed in the plot, the probability of finding a path using this approach quickly increases with the number of sampled subpaths.

Fig. 5 (middle) and (right) show the dependence of the number of bridges (vertices in the graph) and connections between bridges (edges in the graph) on the number of subpaths.

For the multi-goal planning problem, we compare the performance of the Complete Algorithm 1 against the Shortest 1st, Longest 1st, and Combinatorial heuristics. We also compare against the **Approximation** algorithm proposed by C. Chekuri et al. [3] by setting their approximation parameter $i = 3$ (the higher the number the better the approximation and the cost is proportional to $n^{O(i)}$ where n is the number of vertices in the graph). Fig. 6 summaries our findings for solving the multi-goal problem for 3 goals using the environment in Fig. 4. Here we also compare the methods against computing plans independently for each goal and summing their distances (labeled **Independent**). The comparison was done by sampling 200 different configurations of initial and goal locations. We stored the ratios of the total cost for the different multi-goal plans to the cost of the solution obtained from the Complete algorithm. Box plots of these ratios are presented in Fig. 6 using a roadmap generated from 50 subpaths. The Combinatorial approach yields the best results followed by the Shortest 1st heuristic.

We also compared the cost of plans computed using our approach to those obtained by constructing multi-goal, curvature-constrained paths using the rapidly-exploring random tree (RRT) algorithm [14]. In particular, we use a 2D

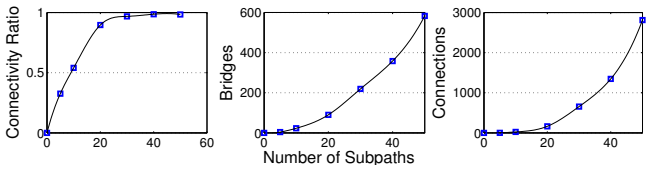


Fig. 5. Dependence of connectivity on the number of subpaths. Each data point is based on the average of 500 independent experiments. (Left) The proportion of runs in which a path connecting the start and goal points is found. (Middle) The number of bridges, i.e. vertices in the roadmap, increases superlinearly. (Right) A similar dependency is observed for the number of connections between bridges, i.e. the edges in the roadmap.

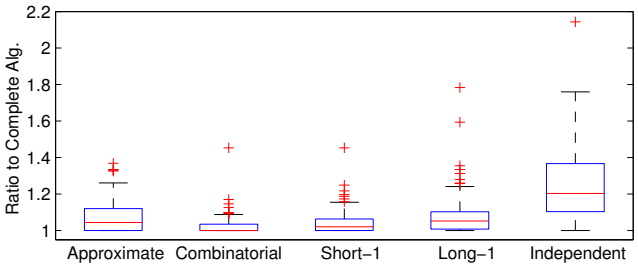


Fig. 6. Comparison of the ratios between the different heuristic and approximation algorithms against the outcome from the Complete Algorithm 1. For each algorithm, we computed ratios for 200 independent experiments involving random sampling of the start points and 3 goal locations. The results correspond to 50 subpaths (about 600 vertices in the graph) while similar results were observed for 20, 30 and 40 subpaths. Ratios close to 1 indicate a good approximation of the Complete Algorithm solution.

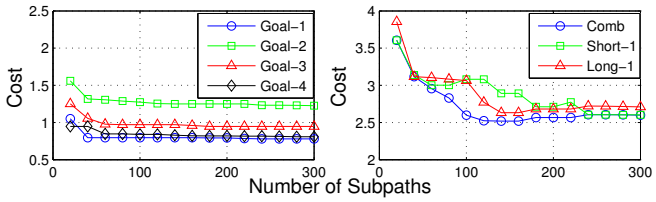


Fig. 7. Cost of the optimal path as a function of the number of subpaths using our approach for the start and goal points in the environment in Fig. 1. Optimal costs for each goal computed individually (left). Cost of heuristics for multi-goal Steiner tree problem (right).

variant of the algorithm proposed by Patil et al. [19] to efficiently compute curvature-constrained paths on a plane. The RRT algorithm has been successfully applied to solving motion planning problems for systems with complex differential constraints, but the method offers no guarantees on optimality. For the test case shown in Fig. 4, the average cost of plans computed using the RRT algorithm across multiple runs was 1.5 times greater than the cost of plans computed using our approach.

Finally, we evaluate the overall optimal performance of the method for single and multi-goal planning using the environment and set of start and goal locations in Fig. 1. As shown in Fig. 7 (left), the cost of an optimal path found by the method decreases toward the optimal solution as the number of sampled subpaths increases. Similarly, Fig. 7 (right) shows how the multi-goal heuristic methods also converge.

VII. Extension to 3D Environments

Our results can be extended to 3D by sampling circles with arbitrary orientation, and then connecting them using a bridge. It is not possible to find a single circular arc that is tangent to two circles in general configuration in 3D. Hence, we will consider using two arcs.

First, we need to model the problem using its forward kinematics. We refer the reader to the text by R. M. Murray et al. [17] for a thorough treatment of reference frames and the use of homogeneous coordinates to represent affine transformations as matrix multiplications. We can model the transformations between two circles tangent to each other by considering two coordinate transformations: one going from the initial circle to an intermediate perpendicular circle, that has its center in the original one; and a similar transformation

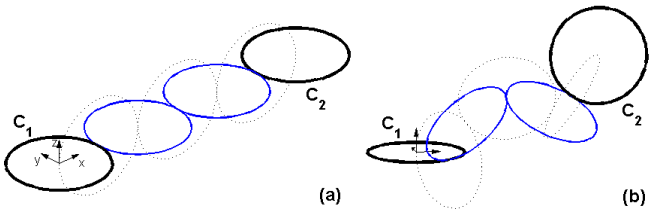


Fig. 8. Connecting two circles with arbitrary location and orientation in 3D using two tangent circles. (a) Base configuration with $\theta_i = 0$. (b) Configuration with arbitrary θ_i values.

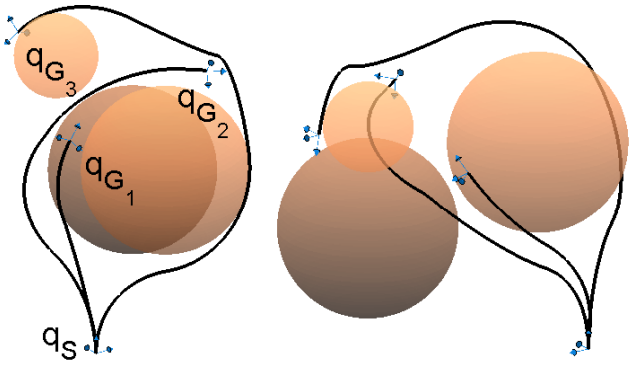


Fig. 9. Results of our multi-goal algorithm applied to an environment with 3 spherical obstacles and 3 goal configurations (point locations and desired orientation). Multiple views of the same plan are shown.

from the intermediate circle to a third circle. This procedure guarantees that the third circle is tangential to the original circle. Hence, we will have a total of 6 transformations if we consider 4 circles. The transformation between adjacent coordinate frames can be expressed as:

$$g_{i-1,i} = \begin{bmatrix} \cos(\theta_i) & 0 & -\sin(\theta_i) & r\cos(\theta_i) \\ \sin(\theta_i) & 0 & \cos(\theta_i) & r\sin(\theta_i) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where θ_i is the angle that specifies the position between circles $i-1$ and i, $g_{j,i} := g_{i,j}^{-1}$ for $i < j$, and $g_{i,j} := g_{i,i+1} \cdots g_{j-1,j}$. We associate with circle C_1 the coordinate frame Ψ_0 and with circle C_2 the frame Ψ_6. Each reference frame is located at the center of the circle with the z-axis perpendular to it. Fig. 8 illustrates the configuration of circles C_1 and C_2 (both in black), the connecting circles (in blue), and the intermediate circles (dotted lines).

For the inverse kinematics problem we are given coordinate frames Ψ_0 and Ψ_6 and would like to compute the angles θ_i. Let us define g as the given transformation from frame Ψ_0 to Ψ_6. Hence, the following equation must be true:

$$g_{0,3}v = g \cdot g_{6,3}v, \quad (3)$$

for $v = v_1 := [0,0,0,1]^\top$ and $v = v_2 := [0,0,1,0]^\top$. Using the value of v_1 ensures that the origin of the frame Ψ_3 is the same when expressed using coordinate frames Ψ_0 and Ψ_6. Using v_2 ensures that the z-axis (which specifies the orientation of the circle) also agrees. We do not check for $g_{0,3} = g \cdot g_{6,3}$ since we are only interested in matching the orientation of the circle and not the exact coordinate frame used to represent it.

The solution of the previous equation is equivalent to finding the minimum to the function:

$$F(\Theta) = ||(I - g_{3,0} \cdot g \cdot g_{6,3})v_1||^2 + ||(I - g_{3,0} \cdot g \cdot g_{6,3})v_2||^2, \quad (4)$$

where I is the 4 by 4 identity matrix. This leads to a straightforward optimization problem.

An example of this approach for 3D configurations is illustrated in Fig. 9, where 1000 circles of radius 0.5 units were sampled over a unit cube with three spherical obstacles

in the interior. The goal configurations q_{G_k} are specified using position and desired orientation.

VIII. Conclusion

We present a new path planning method for robots with curvature constraints on their motion to visit multiple goals in any order. We first introduce a subpath-based roadmap that facilitates computation of curvature-constrained paths that optimize an application-specific metric. This roadmap, which generalizes to both 2D and 3D workspaces, is constructed by sampling circles of bounded curvature and generating feasible transitions between these sampled circles. We then formulate the path planning problem to multiple goals as a Steiner directed tree problem over this roadmap. Since optimally solving the multi-goal planning problem requires exponential time, we propose greedy heuristics to efficiently compute a path that visits multiple goals. We apply the planner in the context of medical needle steering where the needle tip must reach multiple goals in soft tissue, a common requirement for clinical procedures such as biopsies, drug delivery, and brachytherapy cancer treatment. We demonstrate that the proposed heuristics converge to within 5% of the complete, optimal approach and that considering the multi-goal planning problem significantly decreases tissue damage compared to sequential execution of single goal plans.

In future work, we plan to examine formal approximation bounds on the greedy algorithms. For the needle steering application, we plan to investigate the impact of uncertainty in the needle's motion on the optimality of multi-goal plans. We also plan to investigate reducing the risk of needle buckling [21] by removing edges from the roadmap that result in the needle tip moving toward the start location. Finally, we plan to evaluate the performance of our multi-goal planner using both mobile robots as well as steerable needles in artificial phantoms and ex vivo tissues.

IX. Acknowledgement

This work was supported in part by the National Science Foundation under grant # 0937060 to the Computing Research Association for the CIFellows Project and grant # IIS-0905344 and by the National Institutes of Health (NIH) under grant # R21EB011628. The authors also thank members of the needle steering research community for their valuable input.

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] D. A. Anisi, J. Hamberg, and X. Hu, "Nearly time optimal paths for a ground vehicle," *J. Control Theory and Applications*, vol. 1, no. 1, pp. 2–8, 2003.

[3] C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed Steiner tree problems," *Journal of Algorithms*, vol. 33, pp. 73–91, 1999.

[4] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[5] T. Danner and L. E. Kavraki, "Randomized planning for short inspection paths," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2000, pp. 971–976.

[6] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *American J. of Mathematics*, vol. 79, no. 3, pp. 497–516, July 1957.

[7] V. Duindam, J. Xu, R. Alterovitz, S. Sastry, and K. Goldberg, "Three-dimensional motion planning algorithms for steerable needles using inverse kinematics," *Int. J. Robotics Research*, vol. 29, no. 7, pp. 789–800, June 2010.

[8] P. Elinas, "Multi-goal planning for an autonomous blasthole drill," in *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 2009, pp. 342–345.

[9] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Trans. Robotics*, vol. 20, no. 6, pp. 1025–1035, Dec. 2004.

[10] K. Hauser, R. Alterovitz, N. Chentanez, A. Okamura, and K. Goldberg, "Feedback control for steering needles through 3D deformable tissue using helical paths," in *Proc. Robotics: Science and Systems*, June 2009.

[11] D. Hsu, "Randomized single-query motion planning in expansive spaces," Ph.D. dissertation, Computer Science department, Stanford University, 2000.

[12] A. Kelly and B. Nagy, "Reactive nonholonomic trajectory generation via parametric optimal control," *Int. J. Robotics Research*, vol. 22, no. 7, pp. 583–601, July 2002.

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

[14] S. M. LaValle and J. James J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[15] M. Likhachev and D. Ferguson, "Planning long dynamically-feasible maneuvers for autonomous vehicles," *Int. J. Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.

[16] D. Minhas, J. A. Engh, M. M. Fenske, and C. Riviere, "Modeling of needle steering via duty-cycled spinning," in *Proc. Int. Conf. IEEE Engineering In Medicine and Biology Society*, Aug. 2007, pp. 2756–2759.

[17] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.

[18] W. Park, Y. Wang, and G. S. Chirikjian, "The path-of-probability algorithm for steering and feedback control of flexible needles," *Int. J. Robotics Research*, vol. 29, no. 7, pp. 813–830, June 2010.

[19] S. Patil and R. Alterovitz, "Interactive motion planning for steerable needles in 3d environments with obstacles," in *Proc. IEEE RAS/EMBS Int. Conf. Biomedical Robotics and Biomechatronics (BioRob)*, Sept. 2010, pp. 893–899.

[20] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 1, pp. 308–333, March 2009.

[21] K. B. Reed, V. Kallem, R. Alterovitz, K. Goldberg, A. M. Okamura, and N. J. Cowan, "Integrated planning and image-guided control for planar needle steering," in *Proc. IEEE RAS/EMBS Int. Conf. Biomedical Robotics and Biomechatronics (BioRob)*, Oct. 2008, pp. 819–824.

[22] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forward and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[23] M. Saha, T. Roughgarden, J.-C. Latombe, and G. Sánchez-Ante, "Planning tours of robotic arms among partitioned goals," *Int. J. Robotics Research*, vol. 25, no. 3, pp. 207–224, Mar. 2006.

[24] S. N. Spitz and A. A. G. Requicha, "Multiple-goals path planning for coordinate measuring machines," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2000, pp. 2322–2327.

[25] 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, no. 5–6, pp. 509–525, May 2006.