A FOLLOW-THE-LEADER APPROACH TO SERPENTINE ROBOT MOTION PLANNING

Howie Choset Wade Henning
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT. *Serpentine robots offer advantages over traditional mobile robots and robot arms because they have enhanced flexibility and reachability, especially in convoluted environments. These robots are well suited to inspect large space-fairing truss structures such as the future space station and can also be used to inspect the Space Shuttle cargo bay before launch. Serpentine mechanisms offer unique capabilities on Earth to applications such as bridge inspection, search and rescue, surface coating, and minimally invasive surgery.*

The work, described in this paper, will exploit a geometric structure, termed a roadmap, to guide the motions of a serpentine robot in highly convoluted spaces. This approach offers advantages over previous work with serpentine robots because it provides a general mathematical structure that is not mechanism specific, thereby having applications to a large class of problems.

1 INTRODUCTION

We present some preliminary experimental results in the area of *global sensor based planning* for *hyper-redundant robot manipulators.* Recall from (Chirikjian & Burdick, 1990b) that a hyper-redundant mechanism is a kinematically redundant manipulator in which the degree of redundancy is very large or infinite. Such robots are analogous in morphology to tentacles or snakes. *Sensor based planning* incorporates sensory information into some stage of a robotic motion planning, whether it be navigation, locomotion, grasping, etc. Based on information gathered from sensors, global sensor based planning incrementally defines most of the robot's path and constructs a model of the robot's environment. This differs from *local sensor based planning* (Takanashi *et al.* , 1993) which modifies the robot's plan over a span which is short in time or distance, without affecting a world model.

Due to their high degree of articulation, hyper-redundant robots are potentially superior for operations in highly constrained and unusual environments encountered in applications such as inspection of nuclear reactor cores, chemical sampling of buried toxic waste, and medical endoscopy. Mobile hyper-redundant robots can also exploit their many degrees of freedom to offer a novel means for locomotion ((Chirikjian & Burdick, 1991b), (Chirikjian & Burdick, 1995b), (Chirikjian & Burdick, 1995a), (Burdick *et al.* , 1993)) in complex environments. Although the many of degrees of freedom of serpentine robots supply great functionality, they also supply a challenging research problem of how to coordinate all of the actuators in the robot to yield purposeful motion.

The aforementioned applications are characterized by environments which are difficult to precisely model, and thus global sensor based planning schemes are vital to the realistic deployment of hyper-redundant robots in these applications. In this paper, we describe a global sensor based planning heuristic which combines the hyper-redundant kinematic analysis found in (Chirikjian & Burdick, 1990b; Chirikjian & Burdick, 1991b; Chirikjian & Burdick, 1991a; Chirikjian & Burdick, 1994) and (Chirikjian, 1992) with the provably correct sensor based exploratory algorithms found in (Choset & Burdick, 1994; Choset & Burdick, 1995a) and (Choset & Burdick, 1995b). These sensor based motion planning schemes are based on a structure, termed the *Generalized Voronoi Graph.*

2 SNAKE ROBOTS AND APPLICATIONS

Robotics engineers have already produced serpentine robots for various applications. Professor Hirose at the Tokyo Institute of Technology built the first serpentine robot (Figure 7) and studied how such mechanisms can locomote in the plane. This serpentine robot was also mobile. Research serpentine robots in the United States began with the hyper-redundant manipulator built at the California Institute of Technology by Chirikjian and Burdick (Figure 7) who developed some novel end effector placement algorithms for these robots. They also introduced some new snake robot locomotion theory, as well. Takanashi of NEC developed a serpentine robot for the purposes of search and rescue (Figure 7) for survivors in collapsed buildings. In cooperation with Takanashi, engineers at the Jet Propulsion Laboratory (JPL) adapted the NEC design and built the JPL Serpentine Robot for use in space station inspection.

JPL has loaned the Author's group their serpentine robot to further work on motion planning for snake robots. Currently, the JPL Serpentine Robot is being used to investigate bridge inspection. Since serpentine robots can reach more locations at more approach angles in convoluted environments than conventional robots, they provide enhanced capabilities for bridge inspection, which is a costly and dangerous operation. Serpentine robots can reduce the cost and virtually eliminate the danger, while at the same time minimizing traffic delays that are a result of bridge inspection. So, instead of having a person crawl under a bridge or hang from a supported bucket, a serpentine robot can thread its way through the truss structure of a bridge while providing views to the inspector on the roadbed. Although out of scale, Figure 7 demonstrates our vision for robotic bridge inspection.

In medical field, serpentine robots offer high utility for minimally-invasive surgery (Slatkin & Burdick, 1995). This type of operation virtually eliminates the need to open up large portions of the body to perform surgery, allowing for quicker recoveries and hence shorter hospital stays. In 1994, roughly one third of the 21,000,000 operations performed in the United States could have be achieved using existing minimally invasive surgery techniques (Grundgest *et al.* , 1994). However, only 500,000 minimally invasive surgeries were performed; if the maximal 8,000,000 operations had been performed using minimally invasive techniques, then 12% of the GNP for 1993 would have been saved (Grundgest *et al.* , 1994). This estimate does not factor in that workers would sooner return to their jobs because of the significantly shorter recovery time. Current minimimally invasive technology can only reach superficial portions of the body; serpentine robot can extend this reach to allow for minimally invasive operations throughout the body (Slatkin & Burdick, 1995). Figure 7 contains a prototype surgical snake robot built at the California Institute of Technology.

The Pacific Northwest Lab has designed and built a snake robot for the Naval EOD Technology Division in Indian Head, VA. This snake will be used for surgical disarming of bombs. Just as a surgeon can use an arthroscopic surgical instrument to make repairs deep inside the body, without causing damage to the surrounding tissue, a technician can probe the internals of a bomb without accidentally detonating using a serpentine robot. This application requires that the robot also explore a completely unknown three-dimensional environment. See Figure 7. Three-dimensional sensor based planning is necessary for snake robots to assist in the search and rescue of victims in collapsed buildings where access is limited and the geometry of the environment is difficult to evaluate.

3 PRIOR WORK IN MOTION PLANNING

Although applications for serpentine robots are apparent, none of the previous mentioned robots performed in cluttered environments because the control of these devices is challenging. Unlike conventional robots, serpentines possess multiple and redundant degrees of freedom which, while enabling unusual agility, make a serpent robot difficult to manage. Effectively, planning a path for a serpentine robot is a kin to searching a multi-dimensional configuration space between a start and target snake configuration. Raw engineering intuition is not enough; the basic underlying principals of serpentine motion must be considered.

Our approach is to adapt the structure of a rigorous motion planning scheme to a sensor based implementation; one such scheme is based on a geometric structure, termed a *roadmap* (Latombe, 1991). Roadmaps are defined by the following properties: *accessibility*, *connectivity* and *departability*. These properties imply that the planner can construct a path between any two points in a connected component of the robot's free space by first finding a path onto the roadmap (accessibility), traversing the roadmap to the vicinity of the goal (connectivity), and then constructing a path from the roadmap to the goal (departability).

A roadmap called the *Generalized Voronoi Graph* (GVG) has been introduced in (Choset & Burdick, 1994; Choset & Burdick, 1995a) and (Choset & Burdick, 1995b). A key feature of the GVG is that it is one-dimensional in an arbitrary dimensioned space. In the plane, the GVG reduces to the *Generalized Voronoi Diagram* (Ó'Dúnlaing & Yap, 1985) because both structures are one-dimensional. The GVG is amenable to sensor based construction because there already exists a well defined incremental construction technique for the GVG; the incremental construction technique is an adaptation of a numerical curve tracing algorithm (Choset & Burdick, 1995b).

There has been little work devoted explicitly to serpentine motion planning. One approach is based on the definition of *tunnels* through an obstacle field, into which the manipulator "slithers" (Chirikjian & Burdick, 1990a; Chirikjian, 1992). This work, however, does not prescribe any strategy for constructing the tunnels. A local sensor based planning method was reported in (Takanashi *et al.* , 1993) in which the snake robot maintains its end effector location while it locally adapts to a time varying environment. In this method, the entire manipulator is fit within a tunnel and then part of the tunnel is continuously adapted away from any object that becomes unacceptably close.

One of the first global sensor based planning techniques for highly redundant robots is based on a *tactrix* (Reznik & Lumelsky, 1992). This approach, however, assumes that there are perfect sensors on the robot and has not been implemented on a real robot. Finally, none of the aforementioned procedures can be used to map an environment with a robot snake.

In this work, we used the results of the GVG approach to construct tunnels which have a dual purpose: (1) to prescribe a collision-free path for a hyper-redundant manipulator and (2) to serve as a map of an environment that captures all important topological features. These tunnels are constructed in a sensor based fashion.

4 BACKGROUND

In this section, we motivate and introduce the foundation that addresses the control of the many degrees of freedom of serpentines. Initially, consider a conventional robot. Conventional robots can reach a particular location with one or two configurations, i.e., one or two sets of joint parameters can specify the location and orientation of the end effector of a conventional robot. However, in a highly convoluted environment such as truss or rubble pile, a conventional robot will not be able to reach all positions without hitting an object. A serpentine robot can uniquely obtain these end effector locations by flexing around multiple beams and columns. In order to do so the serpentine must progressively negotiate into and around the three-dimensional obstacles to get to its final location. In other words, the serpentine robot is no longer just reaching a desired location, but it is following a *path* to get there. This path is a successive set of configurations which bring the robot from an initial configuration to a final configuration with a desired end effector position and orientation. An important property of this motion is that the body of the serpentine robot follows the path traced by the head.

This work draws from two sub-fields in motion planning: sensor based planning for robots and motion planning for hyper-redundant manipulators, both of which are briefly reviewed in this section.

4.1 Hyper-redundant Manipulator: Inverse Kinematics

The work in (Chirikjian & Burdick, 1990b) uses a *backbone curve* to capture the important macroscopic features of a hyper-redundant robot. This curve is independent of mechanical implementation. Once the backbone curve is determined, the mechanism is "fitted" to the curve. For example, if the snake comprises ten bays, of three degrees of freedom each, then the thirty degree of freedom inverse kinematic problem is reduced to ten three degree of freedom problems, which is significantly simpler and less computationally expensive. Examples of such fitting techniques are reviewed in (Chirikjian & Burdick, 1991b) and (Chirikjian & Burdick, 1994). See Figure 7.

In (Chirikjian & Burdick, 1994), many techniques are introduced for parametrizing the backbone curve. In this paper, we will assume that the Cartesian position of points on a backbone curve can be parametrized in the form:

$$\vec{x}(s,t) = \int_0^s l(\sigma,t)\ \vec{u}(\sigma,t) d\sigma \tag{1}$$

where $s \in [0,1]$ is a parameter measuring distance along the backbone curve at time t. The *backbone curve base* is the point $s = 0$. $\vec{x}(s,t)$ is a vector from the backbone curve base to point s. By convention, $\vec{x}(0,t) = 0$. $\vec{u}(s,t)$ is the unit tangent vector to the curve at s. $l(s,t)$ is the length of the curve tangent and assumes the general form:

$$l(s,t) = 1 + \epsilon(s,t) > 0. \tag{2}$$

$\epsilon(s,t)$ is the *local extensibility* of the manipulator, which expresses how the backbone curve locally expands or contracts relative to a fixed reference state.

The parametrization of Eq. 1 has the following interpretation. The backbone curve is "grown" from the base by propagating the curve forward along the tangent vector, which is varying its direction according to $\vec{u}(s,t)$ and varying its magnitude (or 'growth-rate') according to $l(s,t)$.

In the planar case, the backbone curve is the locus of points:

$$\vec{x}(s,t) = [x_1(s,t),\ x_2(s,t)]^T$$

where

$$x_1(s,t) = \int_0^s l(\sigma,t)\sin\theta(\sigma,t)d\sigma \tag{3}$$

$$x_2(s,t) = \int_0^s l(\sigma,t)\cos\theta(\sigma,t)d\sigma. \tag{4}$$

$\theta(s,t)$ is the angle, measured clockwise, which the tangent to the curve at s makes with the x_2-axis at time t. By convention, $\theta(0) = 0$, and $x_1(0) = x_2(0) = 0$. By comparing equations 1 with equations 3 and 4, it easy to see that $\vec{u}(s,t) = [\sin\theta(s,t),\ \cos\theta(s,t)]^T$ in the planar case. $l(s)$ and $\theta(s)$ are termed "shape functions," as they control they shape of the backbone curve through the forward kinematic relations 3 and 4.

Within the context of this modeling technique, the inverse kinematic problem, or "hyper-redundancy resolution" problem, reduces to the determination of the time varying behavior of backbone curve shape functions that satisfies task requirements. Different hyper-redundancy resolution techniques can be found in (Chirikjian & Burdick, 1990a; Chirikjian & Burdick, 1991b; Chirikjian & Burdick, 1991a; Chirikjian & Burdick, 1994; Chirikjian, 1992). In one approach the backbone curve shape functions are restricted to a "modal form"

$$\theta(s,t) = \sum_{i=1}^{N_\theta} a_i(t)\ \Phi_i(s) \quad l(s,t) = \sum_{i=N_\theta+1}^{N_l} a_i(t)\ \Phi_i(s) \tag{5}$$

where $\Phi_i(s)$ is a "mode function," and $a_i(t)$ is the associated "modal participation factor." $N = N_\theta + N_l$ is the total number of modes, which must equal or exceeds the number of task constraints. Inverse kinematics reduces to the determination of the modal participation factors.

In (Chirikjian & Burdick, 1990a), motion planning of a hyper-redundant manipulator is reduced to the determination of a family of backbone curves which prescribe the collision-free configurations of the robot as the head moves through an environment. Specifically, (Chirikjian & Burdick, 1990a) suggests an approach where portions of the hyper-redundant manipulator backbone curve are constrained in order to avoid obstacles. A constrained portion of the backbone curve is termed a *tunnel*, and in (Chirikjian & Burdick, 1990a) a tunnel is constructed using full knowledge of the world's geometry. However, Chirikjian and Burdick specify the tunnels by hand, and thus in this work we introduce methods for generating the tunnel segments, using a geometric structure called a roadmap.

4.2 Generalized Voronoi Graph

This work uses a roadmap termed the Generalized Voronoi Graph. In m-dimensions, a *Generalized Voronoi Edge* is the set of points equidistant to m obstacles, such that each point on the edge is closer to the m obstacles than any other obstacle. A GVG edge is one dimensional (Choset & Burdick, 1995a) and GVG edges intersect at *Generalized Voronoi Vertices*. A Generalized Voronoi Vertex is a point equidistant to $m+1$ obstacles such that no other obstacle is closer to the $m+1$ obstacles. The Generalized Voronoi Graph is the collection of GVG edges and vertices.

In the planar case, a GVG edge defined by convex obstacles C_i and C_j is denoted $\mathcal{F}_{ij} = \{x \in R^2 : 0 \leq d_i(x) = d_j(x) \leq d_h(x)\ \forall h\}$, where $d_i(x)$ is the distance between x and C_i. Furthermore, in the plane, a GVG vertex defined by the obstacles C_i, C_j and C_k is denoted, $\mathcal{F}_{ijk} = \{x \in R^2 : 0 \leq d_i(x) = d_j(x) = d_k(x) \leq d_h(x)\ \forall h\}$. In the planar case, the GVG reduces to the Generalized Voronoi Diagram, and both structures are connected (Choset & Burdick, 1995a). See Figure 7.

Another key feature of the GVG is that it can be constructed using solely line of sight information (Choset & Burdick, 1995b). The GVG edge incremental construction technique, described in (Choset & Burdick, 1995b), is an adaptation of a numerical curve tracing procedure (Keller, 1987). Assume the robot starts at point on a GVG edge. Using sensor information, the robot computes the tangent to the graph edge and takes a step in a finite direction along the tangent. This step is sometimes called a "prediction step." Then, on a line orthogonal to the tangent, the robot invokes an iterative correction routine which brings the robot back to the GVG edge, as per

$$y^{h+1} = y^h - (\nabla G(y^h))^{-1} G(y^h),$$

where the roots of G define the GVG, $\nabla G(y)$ is the Jacobian (or differential) of G, and y^h is the hth iteration. For a finite step size along the tangent, the robot is guaranteed to converge back to the GVG edge during the correction process. Both the prediction step and correction routine can be done in a sensor based manner (Choset & Burdick, 1995b).

Edge tracing continues until one of two conditions are met: (1) the robot encounters a Generalized Voronoi Vertex, or (2) the robot hits a boundary point. A Generalized Voronoi Vertex is sometimes called a *meet point* because that is where three GVG edges meet. At this point, the robot is equidistant to three obstacles. However, due to sensor noise, it is unreasonable to expect the robot to sense this condition. However, meet points can be robustly detected by looking for an abrupt change in the two nearest obstacles. When this occurs, the robot has passed from one GVG edge to another, i.e., it passed by a meet point. Now, the robot notes the approximate location of the meet point and continues tracing a new GVG edge until it either reaches another meet point or a boundary point. A boundary point is where a GVG edge intersects the boundary. At this point, the robot simply turns around and returns to a previous meet point that has unexplored directions. Once all meet points have all directions explored, the robot terminates its exploration process (Figure 7). The completed structure is set of one dimensional paths which simply and usefully reflect the topology of the environment.

4.3 Clipped GVG

We have developed a novel technique for considering the thickness of the robot without computing the configuration space. Consider a disk or cylinder mobile robot with a non-zero diameter. By setting the detection tolerance for boundary points equal to the robot's radius, we effectively construct the GVG of the configuration space without having to compute the configuration space. This technique takes advantage of the GVG's property of being determined from sensor based range information alone. The resulting structure, known as the Clipped Generalized Voronoi Graph or CGVG, automatically factors the robot's width into the exploration procedure. Boundary points are placed at the mouth of narrow channels, "clipping" the GVG prematurely. The CGVG is the subset of the of the GVG which can be reached for an orientable mechanism of given thickness.

5 SNAKE MOTION PLANNING

The current motion planning approach marries the backbone curve and GVG approaches, described in the previous section. Effectively, the GVG generates a sequence of backbone curves that bring the serpentine robot from a start configuration to a goal. However, it is possible the robot cannot be fitted to back-bone curves generated by the GVG, so the GVG path must be deformed to accommodate the serpentine's mechanical limits.

5.1 GVG Generates Backbone Curves

In the backbone curve paradigm, motion planning is achieved by specifying a continuous series of related backbone curves which bring the robot from an initial configuration to a final configuration. This series, in fact, describes progressively achievable configurations that culminate on a target while avoiding obstacles.

The backbone curve is divided into two components: the free curve and the tunnel curve. The free curve is the portion of the serpentine robot that is "outside" a highly convoluted volume and the tunnel curve is the portion that is "inside." Initially, the serpentine robot entirely fits a free curve. When the serpentine reaches a final configuration inside a highly convoluted environment, then it is largely fit to a tunnel curve.

The GVG of an environment is used to form the tunnel portion of the backbone curve. As the robot moves into the environment, the free curve portion of the backbone diminishes while the tunnel curve grows along the GVG. The result is a continuous parameterization for backbone curves which specify a path that the serpentine robot follows to achieve a goal configuration.

Typically, a serpentine robot is segmented into kinematically sufficient "bays." This means that each bay can arbitrarily position and orient its end-plate with respect to its base. A fitting procedure is used to place and orient each bay along the entire backbone.

5.2 Follow-the-Leader

Motion planning for a serpentine robot is achieved via a follow-the-leader approach. The head of the snake moves along the GVG, while the rest of body follows. Effectively, the GVG generates a sequence of backbone curves that bring the robot from start to goal. Sometimes, this sequence is termed a *backbone curve-path*. We assume that if the head is not already at a boundary point, that it is possible to advance the snake such that the head moves an increment δs farther along the graph. In other words, we advance the robot along the GVG by adding a δs increment from the GVG to the current backbone curve, thereby forming a new backbone. The result of the fitting procedure is later analyzed to check if this assumption is correct, and if so, we update the robot.

After assigning a new head position, the fitting routine proceeds to step back along the GVG, placing joint centers a fixed distance L from the center of the preceding joint. Joint indices start at zero on the head, and count upwards. The the fitting procedure is performed for the entire backbone curve which includes the tunnel curve and the free curve. The free curve can be anything convenient for the snake; for planar snakes, it is generally a spiral storage shape which unwinds as the serpentine enters the workspace. The computing time to fit the snake is a linear function of the total number of links in the mechanism.

5.3 Curve Deformation

The GVG naturally contains a number of sharp angle discontinuities between separate segments arriving at a meet point. The backbone for a given path will have curvature discontinuities at meet points where it switches from one GVG segment to another. Because the snake joint placement represents a finite approximation of the GVG backbone, such corners are cut. This in general smoothes out sharp discontinuities in the backbone and allows the snake to maintain small joint angles (Figure 7).

Unfortunately, the initial guess of the backbone curve-path may contain some configurations to which the mechanism cannot be fitted. Joint limits and finite bay length prevent the serpentine from following all the backbone curves exactly. So, although the backbone curve-path is guaranteed to be a collision free path, a discrete segmented approximation of the curve may intersect obstacles. In this case the backbone curve must be continuously deformed to satisfy joint limits and avoid obstacles. So, once the path from start to goal is determined, we optimize the path subject to a cost function which "massages" the GVG until it is more amenable for snake robots. For example, minimizing a path with respect to curvature makes the path easier for the snake to negotiate.

After a proper deformation process is completed, the resulting set of backbone curves provides a path for the serpentine robot to reach a goal configuration. Otherwise, the procedure can determine with certainty that no such path exists. A deformation procedure that will satisfy all joint limits while optimizing the backbone curve for curvature and length is a current topic of theoretical research.

6 SIMULATION AND DISCUSSION

We have developed a planar simulator that will construct the GVG for a highly convoluted environment, then compute the backbone curves and serpentine configurations. Initially, the entire GVG is generated. This is akin to exploring an unknown environment. The GVG is incrementally constructed using a numerical curve tracing technique as described above. We simulate the sensor range information with a function that computes the absolute distance to convex polygonal and circular obstacles (although the GVG applies to obstacles of any geometry).

The code runs on Sun and Linux Workstations and can compute the GVG for an unstructured two-dimensional environment in typically fewer than five seconds.

There are two basic modes in which path planning can be achieved: (1) exploration and (2) start-goal motion planning. Exploration either generates the full GVG using line of sight information or uses an existing GVG to re-perform a depth-first search of the environment.

There are two sub-modes for start-goal motion planning: (i) goal location is known (maze searching) and (ii) goal location is not known (inspection). The known goal search uses a depth-first search heuristic to reach a goal with known coordinates. Each time the robot encounters a meet point, it chooses a new edge to explore which locally minimizes the robot's distance to the goal. If the robot encounters a boundary point, then it back-tracks to the previous meet point with an unexplored direction, and explores a direction associated with it. This graph search of space continues until the robot is within line of sight of the goal, or all meet points have no unexplored directions associated with them. When this happens, the path planner reports no path from start to goal exists.

If the coordinates of the goal location are not known, then the simulator explores the GVG until the goal is within line of sight. This is akin to searching for trapped survivors in search-and-rescue operations or inspecting a complex structure for defects. We assume that the robot can identify the goal at a distance by sight. See Figure 7.

Alas, the GVG in Figure 7 cannot be negotiated by a real snake robot because joint limits would be violated. However, using an optimization approach that is currently under development, we can optimize the initial path with respect to length, curvature, and safety, to produce a path that is "better" for the snake to follow. See Figure 7.

We have also written a routine to compute the farthest point on the graph from an arbitrary access point. By knowing the farthest point, we can prevent the snake from wrapping unnecessarily around the environment in a depth-first search. This problem occurs for highly cyclic GVG's, when a depth-first search performs most of its activity with the snake wrapped around distant obstacles while exploring segments more easily accessed from other directions. The serpentine can be instructed never to search deeper into the environment than the distance to the farthest point, since a shorter less convoluted route to any such end effector location is known to exist.

As with other computational geometry problems, the construction of the GVG is complicated by the presence of symmetries in the environment, such as those in Figure 7. Meet points typically occur at triple equidistance, where there are three local minima in the robot's visible distance function. Essentially, computational geometers assume no four points can be co-circular; they call this general position. General position is reasonable, in many situations, because it corresponds to stable configurations of objects in the environment. For example, if three points p_1, p_2, and p_3 are equidistant to x, then perturbing any of the three points does not affect the equidistant relationship x has with p_1, p_2, and p_3. However, if x is equidistant to four points p_1, p_2, p_3, and p_4, then if any point p_i is perturbed, x is no longer equidistant to all four points. General position tends to be violated in symmetric environments; assuming general position inside of man-made structures is unreasonable because most man-made structures are symmetric. The planar simulator has been developed for robustness in generating the GVG for such n-way meet points.

Currently, we are extending our simulator work to three-dimensions, where the GVG is the set of points equidistant to three objects. Here, the snake follows this triple equidistant locus to find a path from start to goal.

Finally, the work described in the paper applies to both fixed base and mobile snake robots. For mobile snake robots, the head of the robot simply follows the GVG while body complies to the rest of the GVG or deformed GVG. In fixed based operation, the snake is "coiled" up and unfolds, again with the head following the GVG and the unfolding body complying to the GVG. A hybrid fixed-base-mobile serpentine system can also be employed for applications, such as structure inspection. In this scenario, a large traditional robot arm to provides coarse placement for the serpentine's base. From this base, the serpentine can snake its way through the surrounding structure and provide fine placement for a camera and other sensors situated on its end effector. The serpentine robot guarantees that all locations of one region of the structure are seen, whereas the large robot is used to bring the serpentine to each region. This proposal focuses on the planning and control strategies that achieve the fine mode portion of the inspection procedure.

7 CONCLUSION AND FUTURE WORK

Despite the distinct need for serpentine robots, we believe that they are not used because there were previously no adequate control strategies to direct them. Therefore, this work employs a roadmap termed the Generalized Voronoi graph. Using the Generalized Voronoi Graph for serpentine robot motion planning has several advantages. First, it can be used in a sensor based way, without a priori knowledge of the environment. Second, using a follow-the-leader approach to define backbone curves through the environment, the computing cost associated with a highly redundant manipulator can be greatly reduced. Computational efficiency allows for real time control without full knowledge of the environment. Savings is achieved by not having to perform the manipulator inverse kinematics, not having to compute to configuration space, and by reducing an m dimensional environment to a one dimensional roadmap search space.

An important aspect of this technique is that is can be generalized to three dimensional environments without a significant increase in computation. The fitting procedure for a series of discreet links along a continuous backbone curve is no more computationally expensive than in the two-dimensional case. The time required to compute and update a snake configuration in a three-dimensional environment is of the same order of magnitude as in the two-dimensional case. For the simulation, the computing time required to update a three-dimensional display overwhelms the time used to configure the snake.

Using the GVG already maximizes safety for a given route through the environment by keeping the snake joints as far away from the walls as possible. Optimizing safety is desirable in many real applications such as navigating under the rubble of a collapsed building. Although using the GVG ensures that the backbone curve is placed midway between all obstacles and never intersects with corners, a real snake configuration using links with a high length to width ratio will fit to a coarse approximation of the backbone. This leads to the possibility of the coarse discrete approximation cutting corners. This requires a means to detect obstacle collisions in the discrete approximation and a process for deforming the initial backbone away from the problem area, similar to the method reported in (Takanashi *et al.* , 1993).

A proper solution to the joint limits problem involves the global optimization of the backbone curve with an upper limit on the backbone's curvature. This method is considered superior to the problem of local joint limit handling. We are currently developing a technique using the calculus of variations to optimize the backbone curve for cost functions involving length, curvature, safety, and energy. We plan to use the GVG as a first approximation to the backbone curve, then iteratively deform this seed curve while considering numerous obstacle constraints, and a limited maximum curvature. The first steps of this work was presented in Section 6. A cost function that encodes the exact kinematics of a particular snake is underway. Finally, experiments on the eleven degree of freedom JPL manipulator are underway.

The ultimate goal of this work is sensor based planning for snake robots in unknown environments. This will require retrofitting the JPL snake with a range sensor suite on its head. Once we can access distance to nearby obstacles from such a sensor suite, then the GVG-based method for determining the backbone curve can be implemented in a sensor based fashion. We also intend to put sensors along the body of the serpentine robot to allow for fine-tune adjustments using a local sensor based planner during the follow-the-leader mode.

References

Burdick, J.W., Radford, J., & Chirikjian, G.S. 1993 (May). A 'Sidewinding' Locomotion Gait for Hyper-Redundant Robots. *In: IEEE Int. Conf. on Robotics and Automation.*

Chirikjian, G.S. 1992. *Theory and Applications of Hyper-Redundant Robotic Manipulators.* Ph.D. thesis, California Institute of Technology, Pasadena, CA.

Chirikjian, G.S., & Burdick, J.W. 1990a (May 14-17). An Obstacle Avoidance Algorithm for Hyper-Redundant Manipulators. *Pages 625–631 of: Proc. IEEE Int. Conf. on Robotics and Automation.*

Chirikjian, G.S., & Burdick, J.W. 1990b (Sept. 16-19). Kinematics of Hyper-Redundant Manipulators. *Pages 391–396 of: Proc. ASME Mechanisms Conference.*

Chirikjian, G.S., & Burdick, J.W. 1991a (May 10-15). Kinematics of Hyper-Redundant Locomotion with Applications to Grasping. *In: Proc. IEEE Int. Conf. on Robotics and Automation.*

Chirikjian, G.S., & Burdick, J.W. 1991b (April). Parallel Formulation of the Inverse Kinematics of Modular Hyper-Redundant Manipulators. *Pages 708–713 of: Proc. IEEE Int. Conf. on Robotics and Automation.*

Chirikjian, G.S., & Burdick, J.W. 1994. A Modal Approach to Hyper-Redundant Manipulator Kinematics. *IEEE Trans. on Robotics and Automation*, June.

Chirikjian, G.S., & Burdick, J.W. 1995a. Kinematically Optimal Hyper-Redundant Manipulator Configurations. *IEEE Trans. on Automation and Robotics*, December.

Chirikjian, G.S., & Burdick, J.W. 1995b. The Kinematics of Hyper-Redundant Robotic Locomotion. *IEEE Trans. on Automation and Robotics*, December.

Choset, H., & Burdick, J.W. 1994. Sensor Based Planning and Nonsmooth Analysis. *Pages 3034–3041 of: Proc. IEEE Int. Conf. on Robotics and Automation.*

Choset, H., & Burdick, J.W. 1995a. Sensor Based Planning, Part I: The Generalized Voronoi Graph. *In: Proc. IEEE Int. Conf. on Robotics and Automation.*

Choset, H., & Burdick, J.W. 1995b. Sensor Based Planning, Part II: Incremental Construction of the Generalized Voronoi Graph. *In: Proc. IEEE Int. Conf. on Robotics and Automation.*

Grundgest, W.S., Burdick, J.W., & Slatkin, A.B. 1994. Robotic Endoscopy. *US Patent 5337732*, August.

Keller, H.B. 1987. *Lectures on Numerical Methods in Bifurcation Problems.* Bombay, India: Tata Institute of Fundamental Research.

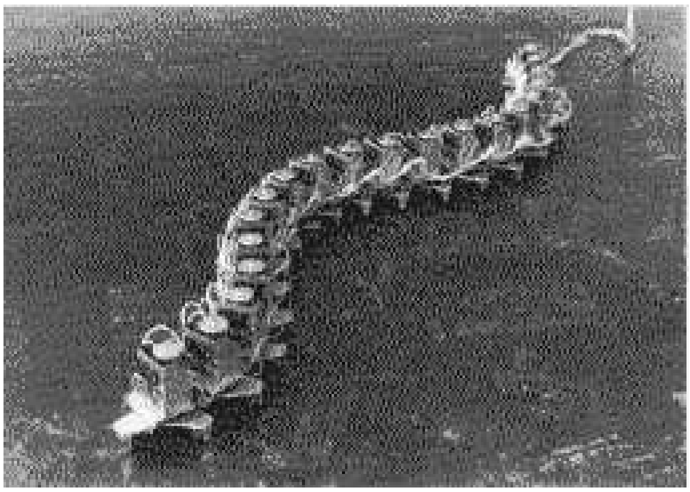
Latombe, J.C. 1991. *Robot Motion Planning.* Boston, MA: Kluwer Academic Publishers.

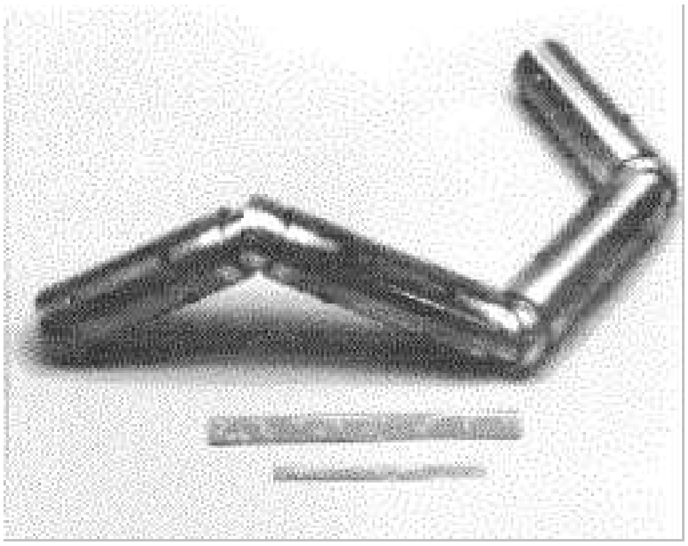
Ó'Dúnlaing, C., & Yap, C.K. 1985. A "Retraction" Method for Planning the Motion of a Disc. *Algorithmica*, **6**, 104–111.

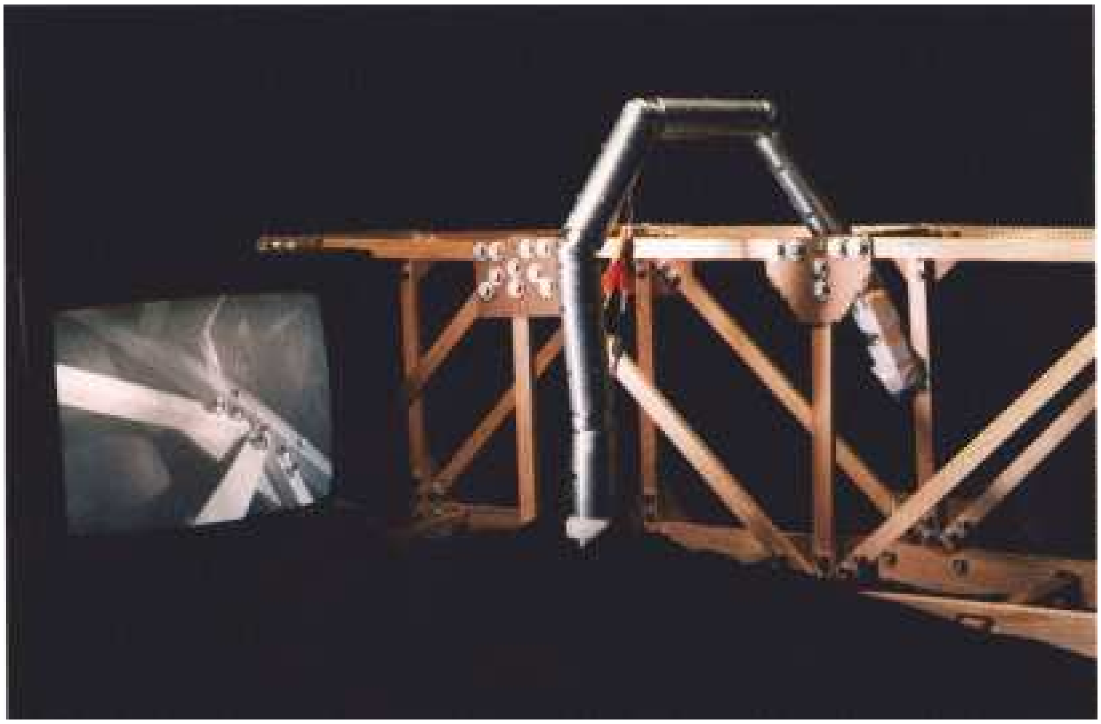
Reznik, D., & Lumelsky, V. 1992. Motion Planning with Uncertainty for Highly Redundant Kinematic Structures I. 'Free Snake' Motion. *In: IEEE/RSJ International Conference on Intelligent Robots and Systems.*

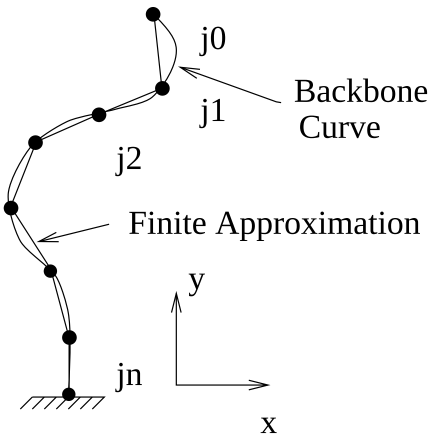
Slatkin, A.B., & Burdick, J.W. 1995 (July). A Minimally Invasive GastroIntestinal Robot. *In: Proceedings. IROS Workshop on Intelligent Robots.*

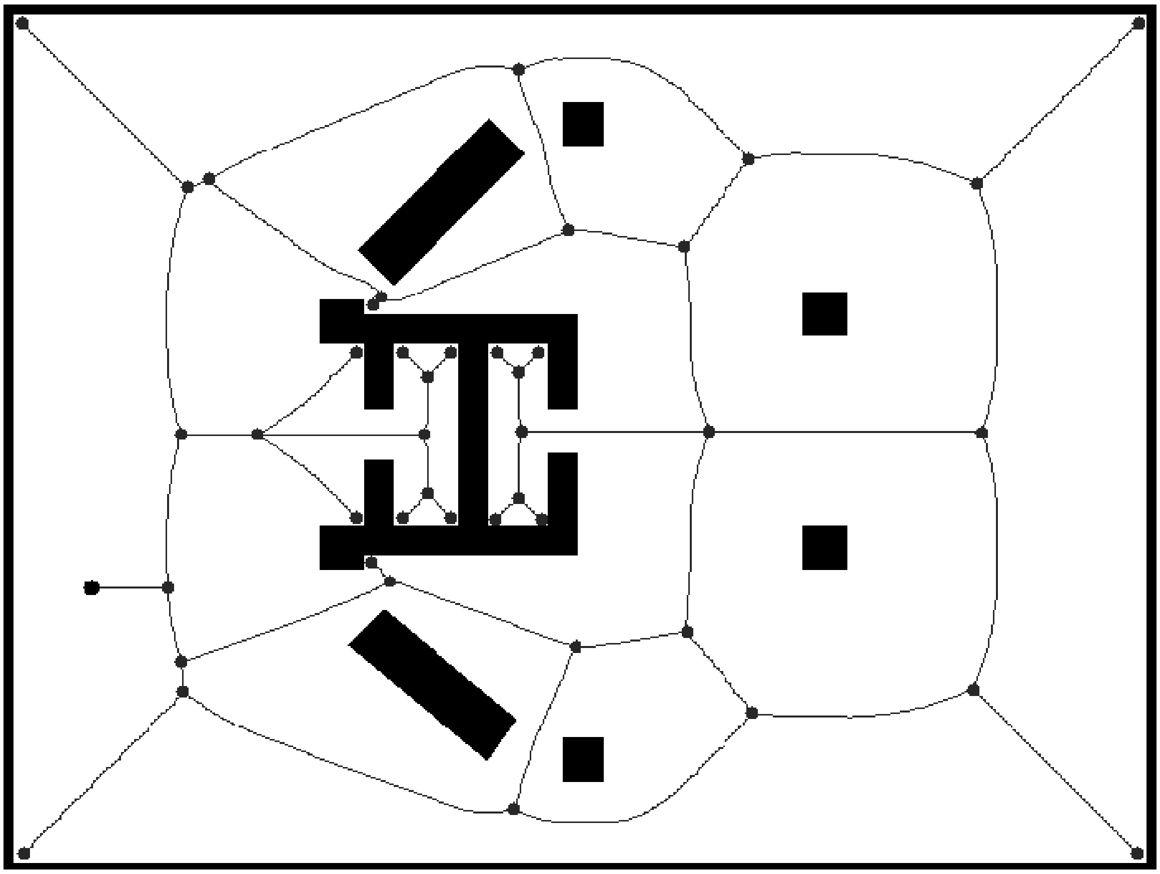
Takanashi, N., Choset, H., & Burdick, J.W. 1993 (July). Local Sensor Based Planning for Hyper-redundant Manipulator. *In: Proc. of IEEE Int. Conf. on Intelligent Robots and Systems.*

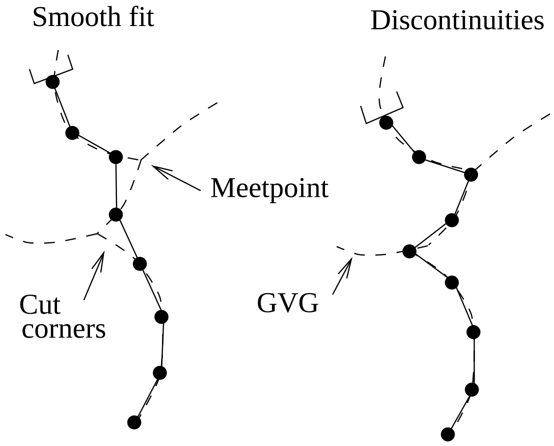


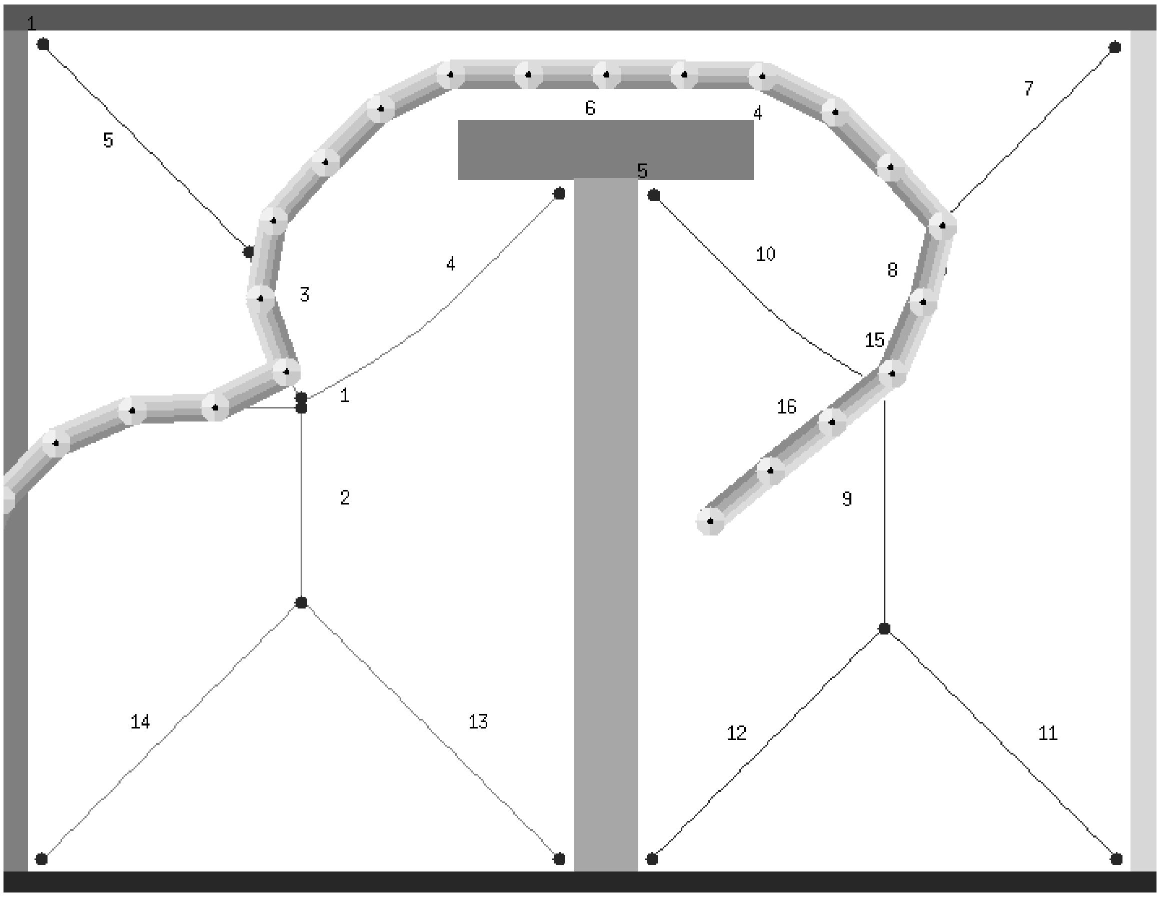


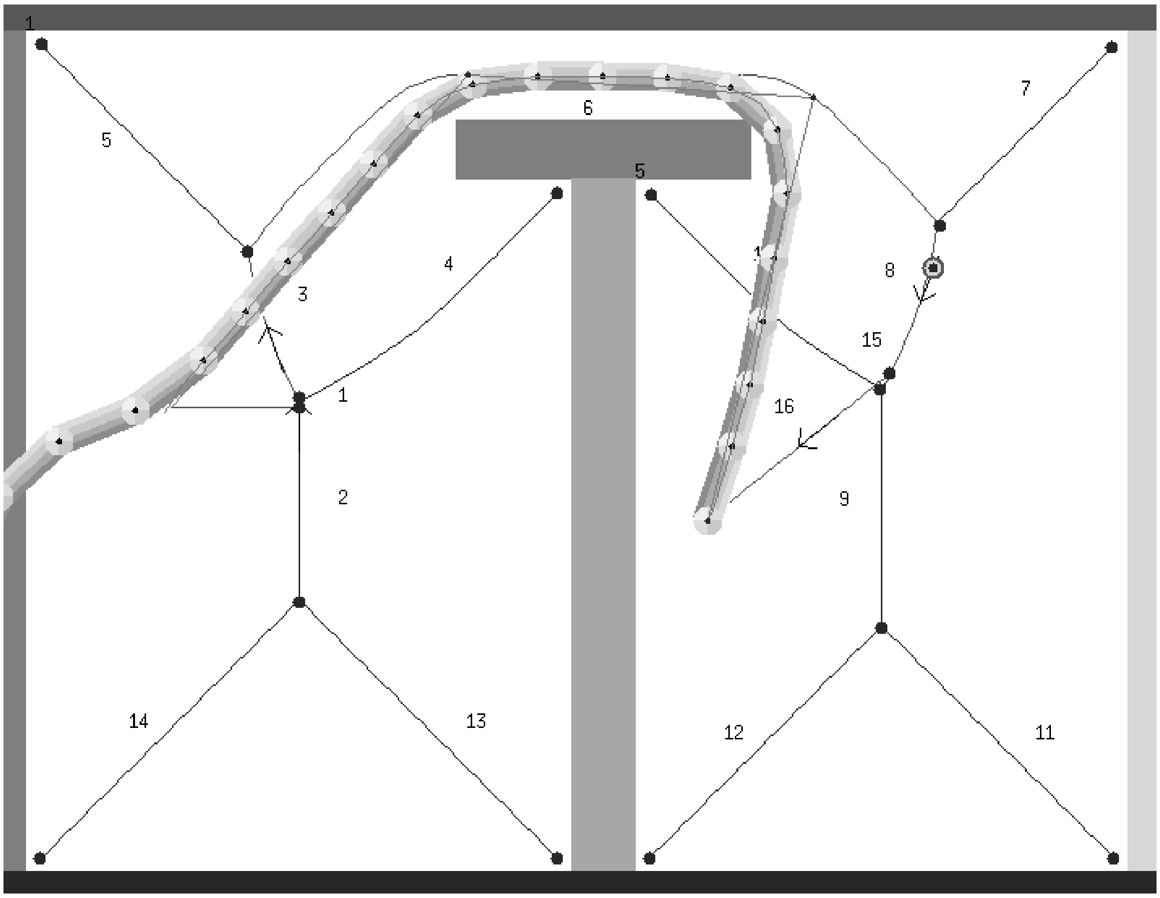



j0
Backbone
Curve
j1
j2
Finite Approximation
y
jn
x




Smooth fit
Discontinuities
Meetpoint
Cut
corners
GVG


1
5
6
4
5
7
4
10
8
3
15
1
16
2
9
14
13
12
11


1
5
6
5
7
4
3
8
15
1
16
2
9
14
13
12
11

Hirose Active Chord

Caltech Snake Robot

NEC Search and Rescue Robot

JPL Serpentine Robot.

JPL Serpentine robot is reaching into a bridge to inspect critical locations. A camera at the end of the robot transmits video images to the monitor on the left.

Caltech Medical Robot Prototype locomotes through the intestines via expanding and contracting the “balloons” on its sections.

Naval EOD Tech Div. Serpentine Robot

Backbone Nomenclature

GVG for a space station cross-section

Effect of fitting procedure on discontinuous backbone

Simulated Serpentine Robot follows GVG

Simulated Serpentine Robot follows GVG