

# Planning Paths for a Flexible Surface Patch

Christopher Holleman    Lydia E. Kavraki    Joe Warren  
 Department of Computer Science, Rice University  
 6100 Main Street, Houston TX 77005  
 {rusty,kavraki,jwarren}@cs.rice.edu

## Abstract

*This paper presents a probabilistic planner capable of finding paths for a flexible surface patch. The planner is based on the Probabilistic Roadmap approach to path planning while the surface patch is modeled as a low degree Bézier surface. We assume that we are dealing with an elastic part and define an approximate energy model for the part. The energy function penalizes excessive shear and bending of the part and we assume that low-energy configurations correspond to reversible elastic deformations of the part. The planner captures the connectivity of a space by building a roadmap, a network of simple paths connecting configurations selected in the space using randomized techniques. We report on the implementation of our planner and show experimental results with examples where the surface patch is required to move through a small hole in its workspace. Our work is a first step towards considering the physical properties of parts when planning paths.*

## 1 Introduction

Path planning has experienced a tremendous growth during the last decade [10]. Currently several planners can deal with complex geometric problems and robots with many degrees of freedom (see [9] for a survey). Despite the vast amount of work on planning, little has been done to consider the physical properties of the robots, such as their flexibility.

In this paper we present a planner that computes paths for a flexible surface patch as a first step in the direction of considering the physical properties of moving objects during path planning. Several important applications motivate our research: in industrial settings flexible objects (e.g. cables) are ignored during planning, although it is well-known that they restrict the feasibility of many paths. In assembly maintainability studies, previous work [5] has focused on planning removal paths for rigid parts, although it is well-known that designers take into account the flexibility of parts to produce compact assemblies [15]. Last but not least, our work may have applications in other domains such as computer generated animation, surgery (i.e. guiding a flexible medical

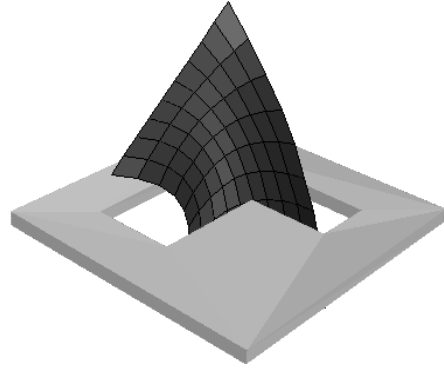


Figure 1: A snapshot along a path produced by f-PRM.

tool inside the human body), and molecular modeling (i.e. docking of two molecules).

We model the surface patch as a low degree tensor product Bézier surface [7] and we plan paths for this surface borrowing ideas from the Probabilistic RoadMap planner (PRM) in [8, 9]. PRM was chosen because it is one of the few planners that can deal efficiently with high dimensional planning problems. Our new planner, which we call f-PRM, proceeds as follows. Initially a large number of configurations are created using appropriate randomization techniques. Configurations that are collision free and have an energy below a user specified threshold are retained as the nodes of a roadmap. The edges of the roadmap are low-energy paths that are found among these configurations by a local planner. Once a sizable roadmap has been produced, f-PRM answers user queries by connecting the initial and the final configuration to the roadmap and then searching the roadmap for a sequence of paths between these nodes. The planner is not restricted to Bézier surfaces; we use this simple model because it allows straightforward manipulation of shape by controlling only a small number of points which are called the Bézier points or control points of the surface. A snapshot along a path produced by f-PRM is given in Figure 1. The surface shown is controlled by 9 points, defining a 27-degrees-of-freedom planning problem.

The surface patch can be regarded as a crude approxi-

mation to a sheet metal or a plastic part which is elastic and is manipulated in a mechanical assembly. At this stage, we use a simple energy model for the part that penalizes extensive stretching and bending of the part. When the energy of a deformation is below a predefined cutoff, the deformation of the part is assumed reversible. Furthermore, we do not account for the tools required to manipulate the surface or the ability of a flexible part to comply with motions of the manipulator by yielding to the contact forces. Research in a similar spirit is done in the area of assembly planning where no manipulators are considered when finding a sequence to assemble a product [22]. At a later stage we expect to incorporate in our planner manipulation constraints and the tools required for manipulation.

Our work blends ideas from planning for high-dimensional problems (PRM) with work in geometric modeling (representations of curves and surfaces) and mechanics/physics (energy models for physical properties). The randomized framework of f-PRM provides an excellent experimental testbed for investigating several novel issues that arise in the context of planning with flexible objects. These include (a) understanding the implications of using different geometric representations for flexible objects, (b) acquiring meaningful energy models for these objects, (c) studying the effect of manipulation on deformable parts, (d) developing algorithms for computing low-energy paths between configurations with different deformations, (e) obtaining efficient methods for collision checking when the shape of the object changes, and (f) studying techniques for improving the overall quality of the paths.

This paper is organized as follows. Section 2 briefly surveys related work. Section 3 outlines the general principles of f-PRM. In Section 4 we discuss our current implementation for the Bézier surface patch. We report experimental results in Section 5. Section 6 the paper.

## 2 Related Work

Recent work on the path planning problem has produced several practical planners for robots that consist of rigid parts (see for example references in [2, 9]). These methods routinely take into account geometric constraints such as joint limits and obstacles, but also constraints arising from kinematics such as nonholonomic velocity constraints due to the rolling without slipping of wheeled mobile robots [11], or constraints over the radius of curvature of a car-like system [12].

Robotics work has also considered several aspects of dealing with physical constraints during planning. The study of dynamics and control have guided the design of modern robots and is an area of active research [17]. Furthermore, the complexity of certain problems that cou-

ple planning and dynamics has been investigated (see for example the work on kinodynamic motion planning [6]).

Deformable robots and moving parts have been studied in the context of manipulation. Robots with flexible links are now being built since they facilitate certain tasks (like hammering a peg into a hole) and their modeling and control is an active area of research (for pointers to current work see [15]). Recent papers consider dynamic analysis of robots with flexible payloads such as two robots manipulating a flexible sheet metal [15] or a vibrating object [18], or solve the task of inserting one end of a flexible wire into a hole while holding the other end [14]. Furthermore, research in snake-like robots has explored “geometric mechanics”, a topic that is relevant to our discussion [16].

Issues related to object deformation have been mostly studied in mechanics, geometric modeling, and graphics. In mechanics there exists a large amount of research on modeling physical properties such as elasticity [3]. In geometric modeling several representations for curves and surfaces have been developed to enable accurate manipulation of shape [7]. In this paper we use tensor product Bézier surface but other models may also be appropriate and this issue is currently under investigation. In graphics physically-based models have been proposed for deformable parts [19, 20]. The use of physical simulation and related optimization techniques as a means of geometric interaction has been applied to animation [21], free-form surface and volume modeling [4], and mechanical design [23]. For a discussion on the dynamic simulation of non-penetrating flexible bodies see [1].

## 3 f-PRM: General Description

f-PRM repeats a basic step until a query is answered or until an predefined amount of time has elapsed. The method can be seen as a single shot method (answering a single user query), or as a method that consists of a preprocessing and a query processing phase. In both cases, the approach is incremental: the constructed roadmap can always be augmented by adding more nodes and more paths, hopefully capturing in greater detail the connectivity of the free configuration space (C-space) and allowing for a larger set of user queries to be answered.

**Preprocessing** f-PRM constructs a roadmap  $R = (V, E)$  by repeating the following basic step.

- *Node Generation* A low-energy deformation is generated uniformly at random. Then  $N$  configurations are obtained by applying a random rigid transformation to the deformation until a collision free configuration is produced. All configurations become nodes of  $R$ .

• *Node Interconnection* Having defined a distance metric in C-space, f-PRM attempts to connect each of the newly generated nodes with its neighbors including both neighbors from the current basic step and neighbors from previous steps. Any produced paths become edges of  $R$ . A simple and deterministic local planner must be used for the interconnections (see discussion in [9]). In particular, there are three main concerns when developing the local planner: (a) the planner should connect two nodes through a sequence of low-energy intermediate configurations avoiding unnecessary energy variations, (b) the planner should be fast as it will be used thousands of times, and (c) the planner should be deterministic to avoid storing the actual paths in  $R$ .

• *Node Enhancement* A probability distribution function is defined over all the roadmap nodes in an attempt to heuristically measure the “difficulty” of the area in which a node lies. f-PRM then selects  $M$  nodes according to that function and generates one new configuration close to each of them. The following heuristic scheme is used: a configuration  $c$  is chosen from the  $N$  configurations in  $R$  with probability (see also [8, 9])

$$Pr(c \text{ is selected}) = w(c).$$

The weights  $w(\cdot)$  above are chosen as in [8]

$$w(c) = \frac{1}{d_c + 1} / \sum_{i=1}^N \frac{1}{d_c + 1},$$

where  $d_c$  is the *degree* of node  $c$ , that is the number of connections that  $c$  has with other nodes. To generate a random node in the neighborhood of  $c$ , we start a random walk from  $c$  and perform a predefined number of reflections. For each reflection, we generate a random direction and move the surface in that direction until the surface collides with an obstacle or a predefined number of steps have been executed. At this point we keep the deformation of the surface constant. The end configuration of the random walk is added to  $R$  and is tried for connection with its neighboring configurations of any deformation in  $R$ . The connected components of  $R$  are updated at the end of node enhancement.

**Query Processing** Given a starting configuration  $s$  and a goal configuration  $g$ , f-PRM attempts to connect them to two nodes  $s'$  and  $g'$  that belong to the same connected component of  $R$ . If successful, the component is searched for a sequence of edges from  $s'$  to  $g'$ .

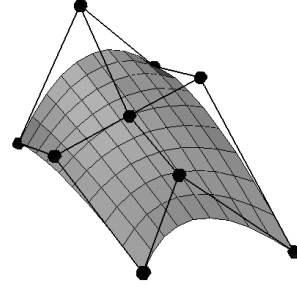


Figure 2: A biquadratic Bézier surface patch with its 9 control points and several isolines.

## 4 f-PRM: The Case of a Tensor Product Bézier Surface Patch

### The Tensor Product Bézier Surface Patch

Bézier surface patches were chosen as the model of a flexible surface in the context of f-PRM. They allow relatively straightforward manipulation of shape while keeping computational costs low when it is necessary to sample points from the surface. A cursory description of Bézier surfaces is offered below (for details see [7]).

A Bézier surface is defined by a control net, made up of a grid of points in 3-space. Let  $P_{ij}$   $i = 0, \dots, n_u, j = 0, \dots, n_v$  be these points, also called the control points or Bézier points. The tensor product approach obtains a point on the surface as

$$S(u, v) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} B_{n_u}^i(u) B_{n_v}^j(v) P_{ij}, \quad (1)$$

where  $u \in [0, 1], v \in [0, 1]$ , and  $B_{n_u}^i(u), B_{n_v}^j(v)$  are the Bernstein polynomials defined by  $B_n^i(t) = \binom{n}{i} t^i (1-t)^{n-i}$ . The surface contains the four corner control points. The rest of the control points are not necessarily on the surface but affect its shape. A simple illustration of a biquadratic surface patch is given in Figure 2.

**Representation of Configurations** The representation of configurations is broken down into three parts

- the load-free (or initial) deformation,
- the deformation, and
- the rigid transformation.

Figure 3(a) shows the object reference frame  $F_A$  for the load-free shape. The load-free shape is input to the program and is represented by a set of vectors  $P_{ij}$  defining the position of its control points in  $F_A$ . At time  $t$  the positions of the control points in  $F_A$  are given by

$$P'_{ij}(t) = P_{ij} + d_{ij}(t)$$

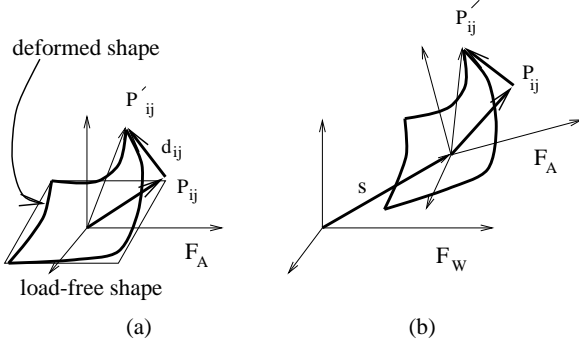


Figure 3: Representation of a configuration.

with  $d_{ij}$  being a displacement function for  $P_{ij}$ . Hence for our purposes we need only keep track of  $d_{ij}$  to represent a deformation. Furthermore, the position of  $P'_{ij}$  with respect to the workspace frame  $F_W$  is given by  $s(t) + R(t)P'_{ij}(t)$ , where  $s(t)$  is the instantaneous translation of  $F_A$  with respect to  $F_W$  and  $R(t)$  is the rotation matrix for the orientation of  $F_A$  relative to  $F_W$  (Figure 3(b)). In our work we implement the above rigid transformation as a translation vector paired with a quaternion. Decoupling deformations and rigid body transformations allows using the same shape at different positions in the workspace and facilitates collision checking as discussed below.

**Energy Model** In order to determine whether a particular deformation should be allowed or not, we devise an energy model that approximates the elastic energy of the plate [3]. The energy model penalizes deformations that induce high curvature, extension, or shear of the surface. We divide the energy into three terms  $E_{curv}$ ,  $E_{area}$ ,  $E_{shear}$  and compute these terms on a  $n \times n$  grid of points sampled evenly across the surface. Let  $x_{ij}$ ,  $i, j = 1, \dots, n$  be the grid points. Let  $x_{ij}^c$ ,  $i, j = 1, \dots, n$  be the corresponding grid points at the load-free shape for referring to properties of that shape. Then

$$\begin{aligned} E_{curv} &= \max_{ij} \{ \sin^2(\text{angle}((x_{ij} - x_{i,j-1}), (x_{i,j+1} - x_{ij}))) \} \\ E_{area} &= \max_i \{ \sum_j |x_{ij} - x_{i,j+1}| - \sum_j |x_{ij}^c - x_{i,j+1}^c| \} \\ E_{shear} &= \max_{ij} \{ \cos^2(\text{angle}((x_{i+1,j} - x_{ij}), (x_{i,j+1} - x_{ij}))) \}. \end{aligned}$$

Once the energy is computed, each component is compared to a predefined cutoff value. In this way we can plan for a path which allows high curvature but relatively little stretching of the surface or vice versa.

**Collision Checking** Collision checking is implemented with the RAPID library [13]. This library takes as input collections of triangles describing the environment and the moving object. In our implementation, Bézier surfaces are approximated by a  $n \times n$  grid of

points evenly sampled across the surface. The obstacles are also decomposed into triangle soups. Once an internal model of the surface and a model of the obstacles have been created by RAPID, a configuration can be queried for collision by specifying a rigid transformation for both models. The creation of an internal model of the surface is expensive compared to the actual collision checks. By keeping the shape of the surface separate from its location in the workspace the internal model for any one deformation can be built once and reused, speeding up collision checking. At this stage, we disallow self-intersections by limiting the curvature of the surface through its energy function.

**Generation of Random Conformations** First a single deformation is generated in compliance with the energy cutoffs. Then random rigid transformations are applied to this deformation to obtain a large number of configurations, which are subsequently checked for collision with the obstacles. The details are given below.

*Deformation* Our presentation here is specific to bi-quadratic Bézier surfaces, but can be extended to higher degree surfaces. Each offset vector  $d_{ij}$  for control point  $P_{ij}$  is created with a random direction and with a magnitude that is chosen uniformly at random from  $[0, l]$ , where  $l$  depends on the position of  $P_{ij}$  in the control net. We bound the magnitude of the offset vectors in an effort to create random shapes close to the load-free shape and avoid excessive deformations of the surface. For the corner points,  $l = 0.5 \max\{\text{length of zero-shape isolines}\}$ , while for the control points on the edges of the net  $l = \|(d_{i-1,j} - d_{i+1,j})\|$  (an additional detail here is that we actually add the vector obtained with this  $l$  to  $(d_{i-1,j} + d_{i+1,j})/2$  to get the offset for  $d_{ij}$ ). For all other control points,  $l$  is computed in a similar way as with the edge control points except that a pair of opposite edge points are used as the neighbors rather than a pair of corner points.

*Rigid Transformation* The translation component  $s$  is simply chosen as a point inside the workspace. We compute the random rotation  $\theta$  by generating a random vector of four values in the range  $[-1, 1]$  and then normalizing the values to obtain a unit quaternion.

**Distance Measure** Let  $c_1, c_2$  be two configurations of the surface. Let  $b_{ij}^{c_1}, b_{ij}^{c_2}$  denote the positions of their control points in  $F_W$ ,  $t_1, t_2$ , the translation vectors in their rigid transformations and  $\theta$  the rotation to obtain  $c_2$  from  $c_1$  as defined by the corresponding quaternions. Then our heuristic distance  $d$  is defined as:

$$d(c_1, c_2) = \|t_1 - t_2\| + |\theta|l_{max} + \max_{i,j} \{ \|b_{ij}^{c_1} - b_{ij}^{c_2}\| \}.$$

In the above,  $l_{max}$  is the maximum length of all the edge isolines of the zero-energy shape divided by  $\sqrt{2}$ . This term approximates the greatest extent of the surface (see discussion in [2]).

**Local Planner** There are two cases for local planning: planning between identical deformations and planning between different deformations. In the first case, the quaternions of the initial and final configuration are compared to find the axis of rotation and angle of rotation needed to get from one orientation to the other. Then a path is constructed by linear interpolation of the translation and rotation vectors. If the deformations of the initial and final object are not the same then the deformations of intermediate configurations must also be linearly interpolated. Since the local planner is deterministic, the actual paths need not be stored in  $R$ ; the edges of  $R$  record the fact that a connection can be achieved using the local planner.

## 5 Experimental Results

We perform experiments with a biquadratic Bézier surface patch in an environment that contains a narrow passage through which the surface should move and whose size is smaller than the size of the surface. We used a  $10 \times 10$  grid to compute the energy of the surface. The code is written in C++ and all the timing results were obtained on an SGI R10000. Our implementation follows closely the description in Section 4 with the exception that enhancement is started after a few basic steps; the number of nodes generated in each basic step is small and our heuristic scheme for enhancement is not very efficient with small roadmaps.

Figure 4 shows an example where the surface needs to bend in order to go through the L-shaped hole in its workspace. The values of the parameters for this problem are:  $N = 200$ ,  $M = 100$ ,  $K = 18$ . The energy cutoffs for the surface were set experimentally in a way that they do not allow the surface to bend or stretch excessively. The random walk consists of a maximum of 10 reflections, each of which can be 100 steps long. We run 10 different times f-PRM changing the value of the random seed generator. The planner reliably solved the problem all 10 times with an average time of 17.6 mins. Snapshots along one of the paths are given in Figure 4. Our running times improve considerably if the size of the surface is reduced. For example, for a surface whose area is 50% smaller f-PRM solves the problem with an average running time of 9.1 min.

A considerably more difficult example is shown in Figure 5: the polyhedral obstacle which obstructs the rectangular hole of the workspace in Figure 4 extends beyond

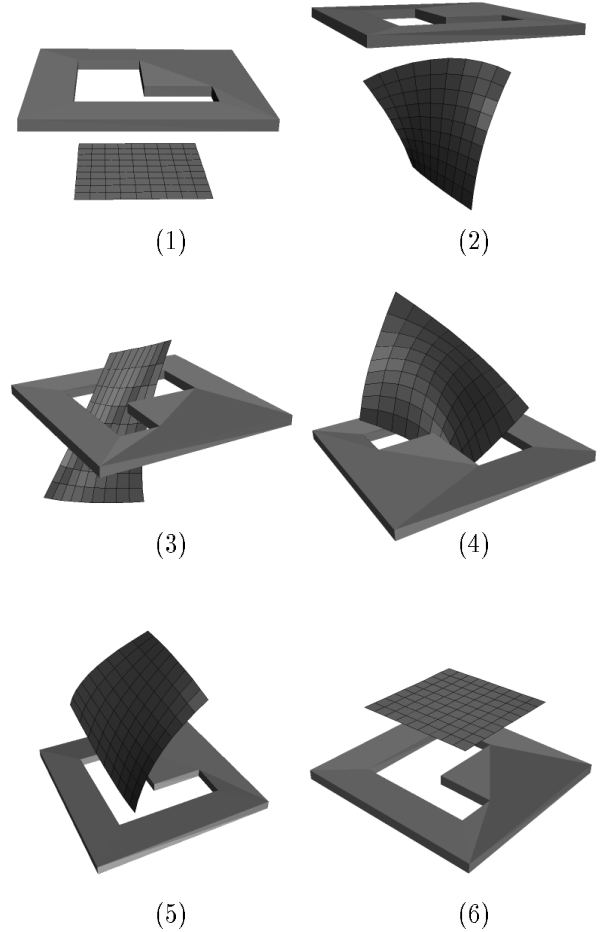


Figure 4: Snapshots along a path computed by f-PRM.

the hole (from below and above). This restricts the deformations with which the surface can go through the hole. It also requires the surface to deform before going through the hole and stay deformed until it is past the vertical obstacle. We solve this problem by keeping the parameters of f-PRM the same, which further attests to the robustness of the planner. We run the planner for 10 different seeds of a random number generator. f-PRM consistently solves the problem with an average running time of 4.7 hours.

## 6 Discussion

This paper discussed a randomized planner, f-PRM, which computes paths for a flexible surface patch. Although we concentrate on the case of a Bézier surface patch that is subject to some energy constraints, our approach is more general. We are currently working towards obtaining a realistic energy model that will reflect the elasticity of the plate. Our work raises several

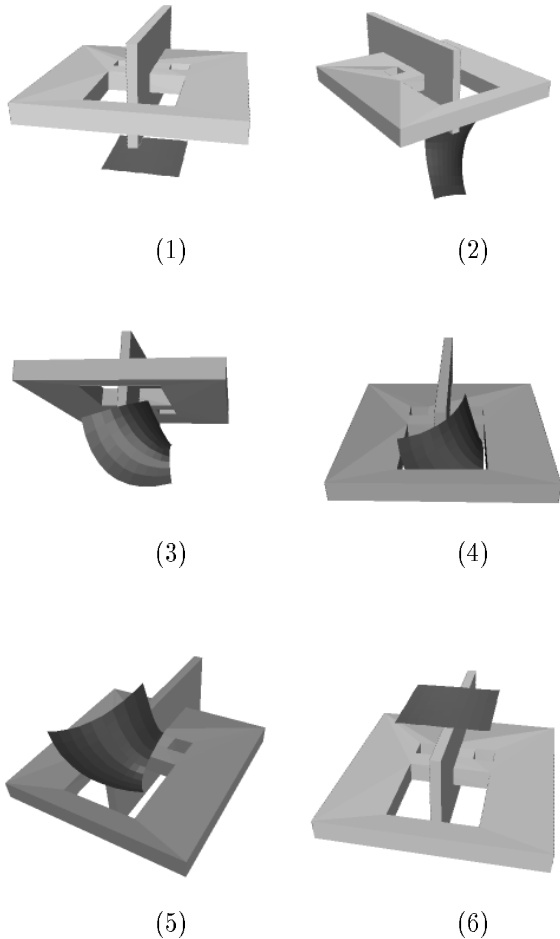


Figure 5: Snapshots along a path computed by f-PRM.

interesting issues that we expect to address in the future. These include: obtaining meaningful energy functions, studying manipulation of deformable objects and the constraints it imposes, and devising methods that smooth our paths or possibly relax the energy of the object along the computed path.

**Acknowledgments** L. Kavraki is partially supported by NSF CAREER Award IRI-970228 and C. Holleman was supported on the RUE portion of the above grant. The authors would like to thank L. Guibas and J.-C. Latombe for their comments.

## References

- [1] D. Baraff and A. Witkin. Dynamic simulation of non-penetrating rigid bodies. *Computer Graphics (SIGGRAPH'92)*, pages 303–308, 1992.
- [2] J. Barraquand and J. Latombe. Robot motion planning: A distributed representation approach. *Int. J. of Robotics Research*, 10:628–649, 1991.
- [3] S. Borg. *Fundamentals of Engineering Elasticity*. World Scientific Publishing Company, Singapore, 1990.
- [4] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. *Computer Graphics (SIGGRAPH'91)*, pages 257–266, 1991.
- [5] H. Chang and T. Li. Assembly maintainability study with motion planning. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 1012–1019, 1995.
- [6] B. Donald and P. Xavier. Provably good approximation algorithms for optimal kinodynamic planning: Robots with decoupled dynamics bounds. *Algorithmica*, 14:443–479, 1995.
- [7] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design*. Academic Press, San Diego, 1988.
- [8] L. Kavraki and J.-C. Latombe. Randomized preprocessing of configuration space for fast path planning. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2138–2145, San Diego, CA, 1994.
- [9] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic roadmaps for fast path planning in high dimensional configuration spaces. *IEEE Tr. on Rob. and Autom.*, 12:566–580, 1996.
- [10] J. Latombe. *Robot Motion Planning*. Kluwer, Boston, MA, 1991.
- [11] J. Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. In *Proc. Intelligent Autonomous Systems*, pages 346–354, 1987.
- [12] J. Laumond, P. Jacobs, M. Taix, and R. Murray. A motion planner for nonholonomic mobile robots. *IEEE Tr. on Rob. and Autom.*, 10:577–593, 1994.
- [13] M. Lin, D. Manocha, J. Cohen, and S. Gottschalk. Collision detection: Algorithms and applications. pages 129–141. A K Peters, Ltd., 1995.
- [14] H. Nakagaki and K. Kitagaki. Study of deformation tasks of a flexible wire. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, Albuquerque, NM, 1997.
- [15] W. Ngugen and J. Mills. Multi-robot control for flexible fixtureless assembly of flexible sheet metal auto body parts. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 2340–2345, Minneapolis, MN, 1996.
- [16] J. Ostrowski and J. Burdick. Gait kinematics for a serpentine robot. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 1294–1299, Minneapolis, MN, 1996.
- [17] M. Spong and M. Viyasagar. *Robot Dynamics and Control*. John Wiley and Sons, NY, 1989.
- [18] D. Sun, X. Shi, and Y. Liu. Modeling and cooperation of two-arm robotic system manipulating a deformable object. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 2346–2351, Albuquerque, NM, 1996.
- [19] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics (SIGGRAPH'87)*, 21(4):205–214, 1987.
- [20] D. Terzopoulos and A. Witkin. Physically based models with rigid and deformable components. *IEEE Computer Graphics and Applications*, pages 41–51, November 1988.
- [21] D. Terzopoulos, A. Witkin, and M. Kass. Energy constraints on deformable models: recovering shape and non-rigid motion. In *Proc. of AAAI 87*, Seattle, 1987.
- [22] R. H. Wilson. *On Geometric Assembly Planning*. PhD thesis, Stanford Univ., March 1992. Stanford Technical Report STAN-CS-92-1416.
- [23] A. Witkin and W. Welch. Fast animation and control of non-rigid structures. *Computer Graphics (SIGGRAPH'90)*, pages 243–252, 1990.