

# Deformable Robot Motion Planning in a Reduced-Dimension Configuration Space

Arthur Mahoney, Joshua Bross, and David Johnson

**Abstract**—Motion planning of deformable objects is challenging due to the high degrees-of-freedom inherent in deformation as well as the computational cost of producing physically accurate deformations. This paper develops a method for fast, physically plausible deformations using a sampling-based planner in a reduced dimensionality configuration space containing position, orientation, and a reduced deformation space produced using principal component analysis. A key element of the approach is use of energy constraints in the configuration space to remove implausible deformations that can be artifacts of the principal component analysis dimension reduction process. This approach is tested on several deforming robot planning tasks, such as a sphere fitting through slots, a bendable bar in a maze, and a deforming plate in tunnels, with collision free paths using plausible deformations found in a few minutes.

## I. INTRODUCTION

Motion planning algorithms have application and utility in a diverse range of fields, such as autonomous vehicles, computer animation and graphics, virtual prototyping, automotive assembly, and biomedicine [12], [16], [19], [20]. The bulk of the research activity has been directed towards the motion planning of rigid bodies. However, recent interest in expanding application of motion planning algorithms to problems in animation, medicine and biochemistry [3], [13], [18] has driven motion planning research for objects that are capable of deforming their shape.

Adding object deformation to the classic motion planning problem increases the problem’s complexity. Fundamentally, the full search space increases exponentially with the degrees of freedom (DOFs) of the modeled robot [7], and deformations can add significantly to these DOFs. Another source of computational complexity is the representation and simulation of the object deformation. If physically realistic deformations are required, which is often the case for biomedical problems [3], mass/spring systems or Finite Element Methods are typically used [14]. Computing iterative solutions for these methods slows down the motion planning process and can result in runtimes on the order of many hours [17]. Problems where semi-realistic deformations suffice can be solved much faster [5], [18], however, the dimensionality of the search can quickly become large to the point of intractability [5].

The goal of this paper is to rapidly produce collision free paths using semi-realistic deformations where the deformations are nearly volume preserving and look physically plausible. The approach presented in this paper takes as input a set of plausible sample deformations of a wire-mesh robot (produced either via simulation or with user knowledge

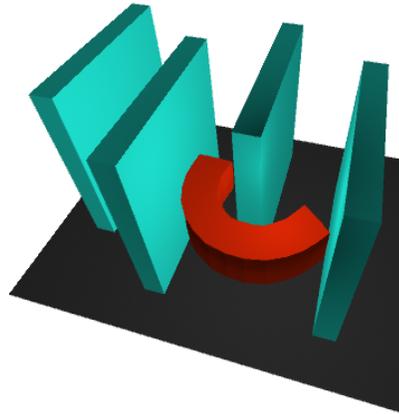


Fig. 1. A puzzle used in this study where a bendable rectangular robot must navigate its way through simple maze of walls. Motion planning problems exist in animation, medicine and biochemistry where the object in motion must deform its shape in order to reach a goal state.

of the robot’s deformation characteristics). From these, a reduced dimensionality representation of the set of sample deformations is created using principal component analysis where a point in the reduced deformation space can be lifted up to the full representation using a linear transformation. Motion planning is then performed by sampling in the configuration space containing both the global motion parameters (translation, rotation, etc.) and the reduced dimension deformation space. Collision detection is performed during motion planning by applying the linear transformation to lift samples in the reduced deformation space to the full deformation space and applying classic collision detection algorithms on the robot under the corresponding deformation. In order to prevent our motion planning method from producing solutions containing unrealistic looking deformations, we apply an energy constraint during the planning process.

The remainder of this paper is organized in the following manner: the next section describes related work in standard and deformable motion planning, Section 3 presents a formal description of deformable motion planning, discusses our application of principal component analysis to reduce the dimension of the deformation space and the motion planner itself. Section 4 presents several examples of deformable motion planning using our method, and finally, the conclusion and follows in Section 5.

## II. BACKGROUND

Before providing more detail on this paper’s approach, it is worthwhile to review related research and provide a

formulation of the deformable motion planning problem. A robot,  $\mathcal{R}$ , has  $d$  degrees of freedom. Some common examples of robots are robots that can translate and rotate in the plane, as in  $\mathcal{R} = \{x, y, \theta\}$ , and robots with  $d$  revolute linkages, where  $\mathcal{R} = \{\theta_1, \dots, \theta_d\}$ . The set of parameters controlling the degrees of freedom has a  $d$ -dimensional domain called the configuration space  $\mathcal{C}$ . In contrast, the workspace, or reachable physical domain of the robot, may be 2- or 3-dimensional. The set of obstacles,  $\mathcal{O}$ , in the workspace where  $\mathcal{O} = \{o_1, \dots, o_n\}$ , maps to a set of obstacles,  $\mathcal{C}_{obs}$  in  $\mathcal{C}$ . A point  $q$  in  $\mathcal{C}$  represents a particular pose, or configuration, of the robot in the workspace, mapped to the workspace by  $\mathcal{R}(q)$ . Thus, points in  $\mathcal{C}$  interior to  $\mathcal{C}_{obs}$  correspond to poses of the robot that would penetrate one or more of the obstacles in the workspace. Points in  $\mathcal{C}$  not in  $\mathcal{C}_{obs}$  are in  $\mathcal{C}_{free}$ .

A deformation,  $d$ , of a wire-mesh robot with  $n$  vertices,  $v = (x_1, y_1, z_1, \dots, x_n, y_n, z_n)$ , can be represented as vector of offsets,  $d = (\delta x_1, \delta y_1, \delta z_1, \dots, \delta x_n, \delta y_n, \delta z_n)$  to the  $n$  vertices of the robot mesh applied at each timestep. The “deformation space” is the set of all deformations of the form  $d = (\delta x_1, \delta y_1, \delta z_1, \dots, \delta x_n, \delta y_n, \delta z_n)$ . If a deformation  $d$  results in the object changing shape in a manner that conflicts with its physical constraints, then  $d \in \mathcal{D}_{obs}$ . In the notation used above, an example rotating, translating, and deforming robot would be defined in the configuration space  $\mathcal{C} = \mathcal{S} \times \mathcal{D}$  where  $\mathcal{S}$  is the set encoding rotation and translation, and  $\mathcal{D}$  is the deformation space. The robot’s configuration in  $\mathcal{C}$ , would be represented by  $\mathcal{R} = \{x, y, \theta, \delta x_1, \delta y_1, \delta z_1, \dots, \delta x_n, \delta y_n, \delta z_n\}$ . This simple definition of mesh deformation was successfully applied to computing grasps of deformable objects in [8]. (Such an approach, however, can quickly yield highly dimensional configuration spaces.)

*The deformable motion planning problem can now be posed as finding a curve  $P(t)$  in  $\mathcal{C} = \mathcal{S} \times \mathcal{D}$  between start and goal configurations  $q_{start}$  and  $q_{goal}$  such that no point in  $P(t)$  is in  $\mathcal{C}_{obs} = \mathcal{S}_{obs} \cup \mathcal{D}_{obs}$ . When  $q_{start}$  and  $q_{goal}$  are not known in advance, then path planners should create a network of curves, or roadmap, which encodes as much as possible of the topology of  $\mathcal{C}_{free}$ .*

Multiple methods have been proposed to perform motion planning of deformable objects using both physically accurate deformations and physically plausible deformations. Using only deformations that look physically plausible during the search path can dramatically increase the performance of the overall search. Several types of physically plausible deformations have been explored. In [5], the authors explore the use of a “bounding box deformation” where the bounding box of an object is deformed to pass through an obstacle space and the object is then applied the same deforming transformation. In the same study, the authors also examine a “geometric deformation” where, if a collision occurs, the colliding polygons of a deformable object are continuously moved until they lie outside of the obstacle. The mass/spring method, which models the deformable object as a collection of vertices connected together by a network of springs, has also proven useful for plausible deformations [18]. In gen-

eral, physical realism comes at the cost of speed, therefore, research is warranted in both areas.

Many existing planners that perform deformable motion planning are based on sampling-based strategies such as the RRT and PRM methods [5], [17], [18]. Sampling-based planners are well suited for deformable motion planning problems because they are typically well behaved in highly dimensional spaces. One interesting approach to motion planning with plausible deformations is utilized by [5], [18]. This method solves the deformable motion planning problem with a PRM in effectively a two-step process. First, a path is found from the initial configuration to the goal configuration allowing collisions within a defined penetration distance. Once the path has been found, each configuration in the bath that produces collision is examined and the deformation necessary to eliminate the collision is determined. In [5], only configurations that produce penetration into the obstacle space within a specified tolerance are added to the roadmap and configurations in the roadmap are connected to each other if the transition between them results in an estimated change of energy below a specified threshold. In [18], on the other hand, the authors produce a possibly colliding trajectory and then simulate deformations using constraint-based dynamics and adjust the path accordingly. (Other interesting research is being performed on motion planning in completely deformable environments [17] and on deformable motion planning for linear objects [13].) In general, these techniques respond to collision by applying deformation to the object until it is no longer colliding. Our approach presented in this paper takes an active approach to deformable motion planning by searching the deformation space for collision free solutions.

### III. METHOD

The space over which our problem is defined includes both the space of position and orientation,  $\mathcal{S}$ , and the space of deformations,  $\mathcal{D}_{\mathcal{F}}$ . The set  $\mathcal{D}_{\mathcal{F}}$  is referred to as the “full deformation space” with dimension  $3n$  where  $n$  is the number of vertices on the robot’s mesh. The configuration space through which a collision free path must be found is  $\mathcal{C}_{\mathcal{F}} = \mathcal{S} \times \mathcal{D}_{\mathcal{F}}$ , which we call the “full configuration space”. Because the dimensionality of  $\mathcal{D}_{\mathcal{F}}$  can become high for even moderately complex models, we utilize a principal component dimension reduction method that reduces the full deformation space to a space of significantly fewer dimensions,  $\mathcal{D}_{\mathcal{R}}$ , which we refer to as the “reduced deformation space”. The resulting configuration space,  $\mathcal{C}_{\mathcal{R}} = \mathcal{S} \times \mathcal{D}_{\mathcal{R}}$ , containing the reduced deformation space is referred to as the “reduced configuration space”.

In brief, our method for deformable motion planning in a reduced configuration space is to first acquire a sample data set of allowable deformations on the object, perform principal component analysis on this set to compute a basis that span the reduced deformation space, then to create a roadmap in the reduced configuration space,  $\mathcal{C}_{\mathcal{R}} = \mathcal{S} \times \mathcal{D}_{\mathcal{R}}$ , using a probabilistic roadmap planner, and finally, to solve for a path in  $\mathcal{C}_{\mathcal{R}}$ . This solution path in  $\mathcal{C}_{\mathcal{R}}$  is then lifted

into the full configuration space,  $\mathcal{C}_{\mathcal{F}}$ , to completely specify the motion of every vertex of the deformable object as it moves through the environment. During the PRM roadmap computation, each sample point is also lifted into  $\mathcal{C}_{\mathcal{F}}$ , where standard collision detection can be performed, but also where energy constraints are computed to maintain plausible deformations of the object. This process can be thought of as a reduced dimension deformable robot planning pipeline as follows (an abstract overview of the pipeline is shown in figure 2):

- 1) Sample the space of allowable deformations through simulation or deformation operators.
- 2) Create a matrix where the deformation samples are the rows and the explicit offsets from the origin of the robot for each vertex in the deforming mesh of the sample are the columns.
- 3) Apply PCA to the matrix to produce a basis for  $\mathcal{D}_{\mathcal{R}}$ .
- 4) Create a linear transformation,  $T : \mathcal{D}_{\mathcal{F}} \rightarrow \mathcal{D}_{\mathcal{R}}$ , defined by the matrix with the basis produced by the previous step as rows. The inverse of  $T$  is  $T' : \mathcal{D}_{\mathcal{R}} \rightarrow \mathcal{D}_{\mathcal{F}}$  which is the transpose of the matrix defining  $T$ .
- 5) Use  $T$  to project the initial and goal configurations of the deforming robot into  $\mathcal{D}_{\mathcal{R}}$ .
- 6) Perform motion planning using a PRM in  $\mathcal{C}_{\mathcal{R}}$ , utilizing  $T'$  to lift sample configurations back up to  $\mathcal{C}_{\mathcal{F}}$  in order to perform collision detection and apply constraints.
- 7) When a solution has been found utilizing  $T'$ , project the configurations of the solution path in  $\mathcal{C}_{\mathcal{R}}$  back up to  $\mathcal{C}_{\mathcal{F}}$  to obtain the solution to the problem.

Our use of principal component analysis and details on the motion planning process, including collision detection and sampling issues, are described in the following subsections.

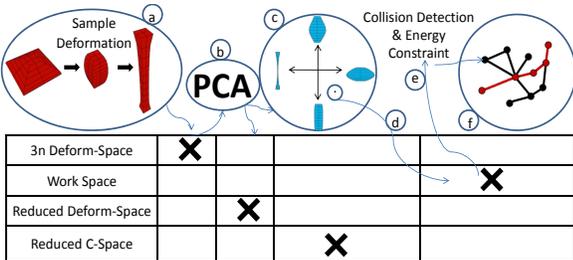


Fig. 2. A graphical depiction of the “pipeline” used by our deformable motion planning method beginning with sample deformation capture, principal component analysis reduction, and motion planning using the sampling-based PRM technique. The dimensional space in which each operation is performed is shown as well.

### A. Reducing the deformation space using PCA

Principle Component Analysis (PCA) has long been used in a wide variety of fields to reduce the complexity of an  $n$ -dimensional data set, containing  $m$  samples, while preserving the intrinsic relationship/connections contained within the data. PCA produces an eigenvector for each column of an  $n \times n$  covariance matrix describing the data set. These orthogonal eigenvectors lie along the direction of the most

variance in the data. The associated eigenvalue,  $\lambda_i$ , of each eigenvector,  $v_i$ , is the variance of the data with respect to  $v_i$  after each point in the data has been orthogonally projected onto  $v_i$ . The dimension of the data set can be reduced from  $n$  dimensions to  $n'$  dimensions by representing each point of data as a linear combination of  $n'$  of the most influential eigenvectors. Principal component analysis has been used in the past to obtain information and reduce the dimensionality of object deformations in the field of computer graphics [1], [2], [9].

In this study, we use PCA to create a linear transformation,  $T$ , that projects deformations from a high dimensional deformation space,  $\mathcal{D}_{\mathcal{F}}$ , down to a drastically smaller deformation space,  $\mathcal{D}_{\mathcal{R}}$ , that is amenable to sampling-based motion planners. To obtain this dimension reducing linear transformation, we first create a data set of  $m$  sample deformations of the form  $d = (\delta x_1, \delta y_1, \delta z_1, \dots, \delta x_n, \delta y_n, \delta z_n)$  in  $\mathcal{D}_{\mathcal{F}}$ . The data set can be populated in many ways. For the experiments described in this paper, we created the data set by collecting  $m$  time captures of the robot as it underwent deformations we defined the robot capable of performing. The data set of deformations is stored in an  $m \times 3n$  matrix,  $M$ , of the following form:

$$M = \begin{bmatrix} \delta x_{1,1} & \delta y_{1,1} & \delta z_{1,1} & \dots & \delta x_{n,1} & \delta y_{n,1} & \delta z_{n,1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \delta x_{1,m} & \delta y_{1,m} & \delta z_{1,m} & \dots & \delta x_{n,m} & \delta y_{n,m} & \delta z_{n,m} \end{bmatrix}$$

A  $3n \times 3n$  covariance matrix is produced from  $M$ , and the resulting set of eigenvectors,  $\{v_1, \dots, v_{3n}\}$  and eigenvalues,  $\{\lambda_1, \dots, \lambda_{3n}\}$  are computed. We select a limited subset of these eigenvectors to become the basis of the  $n'$ -dimensional  $\mathcal{D}_{\mathcal{R}}$  by choosing the  $n'$  eigenvectors with the highest corresponding variance. For data sets of sample deformations that included bending, stretching, compressing, and twisting, the dimensionality of  $\mathcal{D}_{\mathcal{F}}$  could typically be reduced by orders of magnitude. For example, the bending plate puzzle described in the results of this paper used a plate object with 128 vertices yielding a  $3 * 128 = 384$ -dimensional full deformation space  $\mathcal{D}_{\mathcal{F}}$  that was reduced to a 5-dimensional basis using PCA. Figure 3 shows the variance of the first ten eigenvectors produced by PCA for the plate puzzle. The first five eigenvectors were selected to be the basis for the  $\mathcal{D}_{\mathcal{F}}$  space. If the sample deformations are highly non-linear, the intrinsically linear PCA reduction method will fail to reduce the dimensions with minimal loss of information. In this case, non-linear reduction schemes must be used.

The linear transformation,  $T$ , that performs the dimensional reduction on deformation vectors in  $\mathcal{D}_{\mathcal{F}}$  is produced using the  $n'$  eigenvectors computed with PCA. The matrix representing the transformation  $T$  is of size  $n' \times n$  and is simply the chosen  $n'$  eigenvectors in the rows. Once the reduction transformation is obtained, it is trivial to compute the “inverse” transformation,  $T'$ , that lifts the lower dimensional configurations in the  $\mathcal{D}_{\mathcal{R}}$  space back into the full deformation space. The matrix representing the  $T'$  is the  $n \times n'$  transpose of the matrix representing  $T$ . A point in  $\mathcal{D}_{\mathcal{R}}$  represents a deformation of the object after it has been lifted up into  $\mathcal{D}_{\mathcal{F}}$  using  $T'$ . Motion planning is then performed in

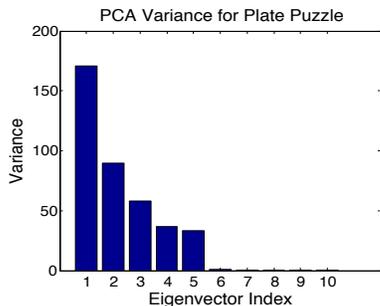


Fig. 3. This figure shows the variance of the sample deformation data along the first 10 eigenvectors produced by PCA for the plate puzzle (see results). The first five with the highest variance were selected to span the reduced deformation space, reducing the deformation space from 384 dimensions to five deformations.

$C_R$  containing position, orientation, and components in  $D_R$  with samples in  $D_R$  being lifted up into  $D_F$  to test collision and deformation constraints.

### B. PRM using reduced deformation space

The Probabilistic Roadmap Method is a random sampling-based technique that is well suited for deformable motion planning because it is often used to efficiently search in highly-dimensional spaces [11]. The PRM technique explores a given configuration space in two phases. In the first phase, the construction phase, a roadmap (graph) is built that approximates the motions that can be made in the space. This is performed by randomly sampling configurations out of the obstacle free space and connecting it to the nearest of its neighboring configurations already sampled. An edge is connected between two sampled configurations if the robot can freely travel between the two without entering the obstacle space. In the second phase, the query phase,  $A^*$  graph search algorithm is used to find the path of travel from the initial configuration to the final configuration through the constructed roadmap. The construction and query phases are repeated until the query is able to return a solution.

The PRM for deformable motion planning takes as input,  $C_R$ , the start and goal configurations of the problem projected into  $C_R$ , and the inverse of the reduction transformation,  $T'$ , that lifts a configuration in  $D_R$  back into  $D_F$ . During the roadmap construction, configurations are randomly sampled out of  $C_R$  and connected to their neighbors with edges. During this phase, the configurations along the edges are tested for collision. Collision detection is performed on the reduced configuration  $x = (s, d_r) \in C_R$  by lifting  $x$  into  $C_F$  using  $y = (s, T'(d_r))$ , updating the coordinates of the vertices in the object mesh from the offsets of the deformation space components in  $y$ , applying the necessary translations and rotations described by  $s$ , and then performing collision detection and constraint satisfaction tests. If collision is detected or a constraint is violated, then the edge in  $C_R$  is tossed out. During the query phase, if the goal has been reached, then the path in  $C_R$  is returned. Lifting the configurations on the path in  $C_R$  into  $C_F$  using the same

method for collision detection results in the solution path for the unreduced problem. (Note: the collision detection methods used in this study do not test for self-collision.)

By representing the space of deformations as a  $3n$ -dimensional configuration space, we are drastically increasing the dimensionality of the motion planning problem. Although the dimensionality can generally be significantly reduced using PCA, other challenges can be inadvertently added to the problem. These include the introduction of narrow passages and artifacts caused by naively sampling deformations from  $D_R$  and lifting them up to  $D_F$ .

1) *Narrow passageways in the deformation space:* Often times it is the case that an object must deform its shape in a very specific way in order to pass through the obstacle space (see the stretching sphere and deforming plate puzzles in the next section, for example). If the specific deformation can only be attained by sampling within  $D_R$  and the PRM must produce a roadmap connection through the area, then a narrow passage in the deformation space is effectively added to the problem. Randomized sampling-based motion planners notoriously have difficulty searching for paths through narrow passages where the probability of sampling out of the passage is very small in comparison to the probability of sampling in other regions of the configuration space. Fortunately, several novel methods have been produced to improve their effectiveness in this situation [4], [10], [21]. We have chosen to use the simple Gaussian Sampling Strategy [6].

2) *Artifacts introduced by sampling in a lower dimension space:* When a configuration is sampled from  $C_R$ , the projection of the components corresponding to  $D_R$  back up to  $D_F$  won't necessarily represent the object under a plausible deformation. Because the dimension of  $D_R$  is typically much smaller than  $D_F$ , an amount of extraneous "information" must be added in some way to a deformation in  $D_R$  in order to transform it to a deformation of much higher dimension in  $D_F$ . Although, the effect of the extraneous "information" is reduced using transformations produced by PCA, lifting some randomly sampled deformations to higher dimension can produce unwanted artifacts that result in unrealistic deformations. This problem can be solved by either increasing the dimension of  $D_R$ , or by applying constraints.

In this study, we have added a simple elastic energy constraint to the motion planning process in order to reduce the affect of projection artifacts. During the roadmap construction phase, when a sample configuration,  $x$ , in  $C_R$  is lifted up to  $C_F$  for collision detection, the elastic potential energy of the robot under the deformation specified by the deformation components of  $x$  is computed. If the energy falls outside the bounds of an acceptable range, then it is deemed to lie within the obstacle space of the problem. The potential elastic energy of a configuration,  $y \in C_F$ , is denoted by  $E(y)$ .  $E(y)$  is calculated by iterating over the edges,  $e_j$ , of the vertex mesh of the robot under deformation specified by  $y$  and summing the elastic potential energy of each edge

as if every edge were a spring:

$$E(c) = \sum_{\forall j} \frac{1}{2} k \|e_j - l_j\|^2$$

where  $e_j$  is the length of the  $j^{\text{th}}$  edge after deformation,  $l_j$  is the length of the  $j^{\text{th}}$  edge with no deformation, and  $k$  is the spring constant. Stretching and compressing the edges beyond their undeformed lengths increases the overall energy of the object. Applying the simple elastic potential energy constraint removes configurations from the roadmap that produce extreme, unrealistic deformations. Other constraints such as object volume and other types of energy measures can be easily added to the motion planner. (Additionally, we have found that by sampling configurations in  $D_R$  within the axis-aligned bounding box of each frame of the input deformation projected from  $D_F$  down to  $D_R$ , fewer configurations are found to be in the obstacle space due to the energy constraint.)

#### IV. RESULTS

To examine the effectiveness of our method for deformable motion planning, we have implemented our technique and benchmarked it on three puzzles. The puzzles include a wall puzzle where a deformable sphere must fit through several holes of varying shape in four walls (adapted from the Wall Puzzle developed by the Parasol Motion Planning group [15]), a very simple maze through which a deformable rectangular block must fit, and a problem where a deformable, flat, rectangular plate must fit a series of pipes. Our experiments were performed on a system with 8 Gb of RAM and a 2.2 Ghz processor. Each puzzle was executed 100 times and the average results are shown in figures 4 and 5.

Puzzle	Tot. Time	% Collision	% Projection
<i>Bending Rectangle</i>	99.0s	26.1%	3.9%
<i>Stretching Sphere</i>	272.7s	87.6%	9.7%
<i>Bending Sheet</i>	238.7s	67.4%	3.1%

Fig. 4. The average total time required to solve each puzzle along with the percentage of time spent performing collision detection and the percentage of time spent projecting configurations from the reduced configuration space to the full configuration space.

Puzzle	Tot. Nodes	Tot. Sampled	Tot. Edges
<i>Bending Rectangle</i>	4,664	297,510	15,675
<i>Stretching Sphere</i>	8,112	157,000	26,814
<i>Bending Sheet</i>	14,217	499,330	53,527

Fig. 5. The average number of nodes required to solve each puzzle along with the number of configurations sampled and the total number of edges in the PRM roadmap.

##### A. Bending Rectangle Maze

The bending rectangle maze takes a rectangle capable of bending in one axis through a very simple maze consisting of a set of offset serial walls. The rectangle robot must bend, transform, and rotate in order to get through each

constriction of the maze. There are 122 vertices in the robot's vertex mesh, and with the robot capable of translating in two DOFs and rotating in one plane, the high-dimensional configuration space contains a total of 368 dimensions. We were able to accurately represent the bending deformation using 3 dimensions resulting in a reduced configuration space of 6 dimensions in total. Figure 6 shows samples along an example solution path for the bendable robot through the maze.

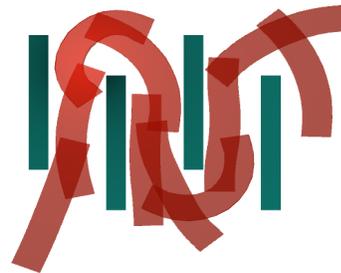


Fig. 6. An example solution to the path for the bending rectangle maze where a rectangle robot is required to bend through a simple maze.

##### B. Stretching Sphere Puzzle

The stretching sphere wall puzzle requires a sphere to alter its shape in order to pass through a set of four walls, with each wall containing a differently sized and shaped hole. The sphere can undergo a compression and stretch deformation that results in the sphere becoming flat or elongated respectively. The triangle mesh representing the deforming sphere contains 382 vertices, yielding a 1146-dimension deformation space. In the workspace, the sphere was permitted to move in the  $x$ ,  $y$ , and  $z$  directions without rotation. This results in a full configuration space containing 1149 dimensions in total. Using the PCA described in the preceding section, we decreased the dimensionality of the deformation space down to a reduced deformation space containing only three dimensions with minimal error. The reduced configuration space in which motion planning is performed, therefore, contained only 6 dimensions. Figure 7 shows an example solution path to the problem with three walls.

##### C. Deforming Plate Puzzle

The deforming sheet puzzle requires a flattened rectangular robot, capable of twisting and bending along its longitudinal and lateral axis, to navigate itself through a series of pipes to the goal. Each pipe of the puzzle requires the robot to bend in a different direction without rotation and translation was restricted exclusively to the  $x$ ,  $y$ -plane bisecting the pipe puzzle. The initial sample deformation included motions that produced the longitudinal bend, lateral bend, and the twist. Because the initial deformation data contained multiple, unique, deformations, the variance of the data was much higher than the preceding two puzzles. We chose to encode the reduced deformation using 9 dimensions and because

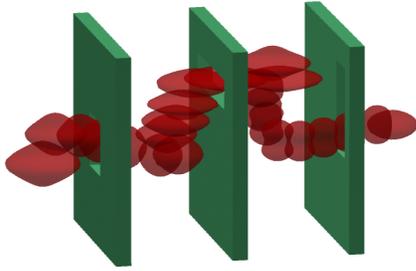


Fig. 7. An example solution to the path for the stretching sphere puzzle (with three walls) where a rectangle robot is required to stretch and compress through a set of serial walls (modified from the serial wall puzzle made available to the public by the Parasol Motion Planning group [15]).

no rotation was allowed and translation was limited to the  $x, y$ -plane, the total dimensions of the configuration space is 11. This puzzle illustrates how multiple deformations can be captured in the initial sample deformation (at the beginning of the pipeline), and how sampling deformations out of  $D_R$  can produce deformations with shape taken from each of the deformations given in the initial sample. Figure 8 shows samples taken from the solution path and shows deformations using combinations of bends along both axis and twists.

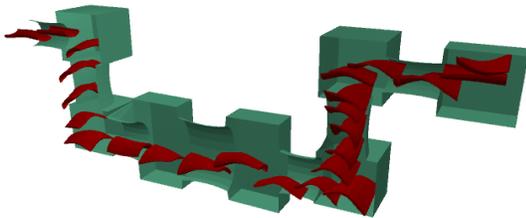


Fig. 8. An example of a solution path to the deforming plate puzzle. The deformations along the path are combinations of twists and bends along the longitudinal and lateral axis.

## V. CONCLUSION

The reduced dimension configuration space described in this research addresses the challenge of motion planning for a deformable robot under physically plausible deformations in a way consistent with global robot degrees of freedom such as translation and rotation. Thus, planning approaches developed for efficient planning in high-dimensional configuration spaces apply directly to deformation planning as well. The linear PCA dimension reduction preprocess step described in this study performs well for the example problems presented, however, several challenges must be overcome in order to effectively produce good solutions to given deformable motion planning problems. Namely, including the reduced deformation space in the configuration space of the robot

can unintentionally add narrow passages to the problem, and straightforward sampling in the reduced configuration space can yield implausible deformations of the reconstructed robot. Our results show that existing sampling strategies can mitigate problems caused by narrow passages in the deformation space and rather than using a more complex dimension reduction scheme or additional deformation dimensions, we found that rejecting samples based on energy constraints was effective in retaining reasonable deformations. The method presented in this paper will be further developed for research in planning problems for animation and medicine.

## REFERENCES

- [1] J. W. A. Pentland. Good vibrations: Modal dynamics for graphics and animation. In *SIGGRAPH*, pages 215–222, 1989.
- [2] M. Alexa and W. Muller. Representing animations by principal components. In *European Association for Computer Graphics*, pages 411–418, 2000.
- [3] R. Alterovitz, K. Goldberg, J. Pouliot, and I. Hsu. Sensorless motion planning for medical needle insertion in deformable tissues. *IEEE Trans. Information Technology in Biomedicine*, 13(2):217–225, 2009.
- [4] N. Amato, O. Bayazit, L. Dale, C. Jones, and D. Vallejo. OBPRM: an obstacle-based PRM for 3D workspaces. In *Proc. Int'l. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1998.
- [5] O. Bayazit, J. Lien, and N. Amato. Probabilistic roadmap motion planning for deformable objects. In *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pages 2126–2133, 2002.
- [6] V. Boor, M. Overmars, and A. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pages 1018–1023, 1999.
- [7] J. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA, USA, 1988.
- [8] K. Gopalakrishnan and K. Goldberg. D-space and deform closure: a framework for holding deformable parts. In *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pages 345–350, 2004.
- [9] K. Hauser, C. Shen, and J. O'Brien. Interactive deformation using modal analysis with constraints. In *Graphics Interface*, pages 247–256, 2003.
- [10] D. Hsu, T. Jiang, J. Reif, and Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pages 4420–4426, 2003.
- [11] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.
- [12] S. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, MA, 2006.
- [13] M. Moll and L. Kavraki. Path planning for deformable linear objects. *IEEE Trans. on Robotics*, 22(4):625–636, 2006.
- [14] A. Nealen, M. Muller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836, 2006.
- [15] Parasol Laboratory. Motion planning benchmarks, Sept. 2009. <http://parasol.tamu.edu/groups/amatogroup/benchmarks/mp/>.
- [16] J. Pettre, T. Simeon, and J. Laumond. Planning human walk in virtual environments. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 3048–3053, 2002.
- [17] S. Rodriguez, J. Lien, and N. Amato. Planning motion in completely deformable environments. In *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pages 2466–2471, 2006.
- [18] G. Russell, M. Ling, and D. Manocha. Constraint-based motion planning of deformable robots. In *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pages 1046–1053, 2005.
- [19] A. Schweikard, J. Adler, and J. Latombe. Motion planning in stereotaxic radiosurgery. *IEEE Transactions on Robotics and Automation*, 9(6):764–774, 1993.
- [20] G. Song and N. Amato. Using motion planning to study protein folding pathways. In *Proc. of Int'l Conf. on Computational Biology*, pages 287–296, 2001.
- [21] S. Wilmarth, N. Amato, and P. Stiller. MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pages 1024–1031, 1998.