Disconnection Proofs for Motion Planning

Julien Basch   Leonidas J. Guibas   David Hsu   An Thai Nguyen
Computer Science Department
Stanford University
Stanford, CA 94305, USA

Abstract

Probabilistic road-map (PRM) planners have shown great promise in attacking motion planning problems with many degrees of freedom that were previously infeasible. Yet when such a planner fails to find a path, it is not clear that no path exists, or that the planner simply did not sample adequately or intelligently the free part of the configuration space. We propose to attack the motion planning problem from the other end, focussing on *disconnection proofs*, or proofs showing that there exists no solution to the posed motion planning problem. Just as PRM planners avoid generating a complete description of the configuration space, our disconnection provers search for certain special classes of proofs that are compact and easy to find when the motion planning problem is 'obviously impossible,' again avoiding complex geometric and combinatorial calculations. We demonstrate such a prover in action for a simple, yet still realistic, motion planning problem. Failure of the prover suggests key milestones, or configurations of the robot that can then be passed on and used by a PRM planner. Thus by hitting the motion planning problem from both ends, we hope to resolve the existence of a path, except in truly delicate border-line situations.

1 Introduction

Motion planning, the discovery of an obstacle-avoiding path between two given robot configurations, has been a key problem in robotics for many years. A variety of methods, based on ideas such as road-maps, cell decompositions, and potential fields, have been proposed (see [Lat91] for a survey), but the majority of them can only handle robots with few degrees of freedom (DoF). The difficulty of this classic question stems from the complexity of the configuration space that must be searched in order to find a path. Combinatorially, the number of features in such a space can be doubly exponential in the number of DoFs of the robot. This makes a full construction or exploration of the space impractical, except for simple robots with very few DoFs. In order to address this complexity, researchers have focussed on ways to discover the connectivity of the free part of the configuration space, in which all paths of interest lie, without explicitly constructing the space itself. Examples of such approaches are the roadmap planner of Canny [Can88] and the probabilistic roadmap (PRM) method introduced by Kavraki *et al.* [KŠLO96].

In a PRM planner, the configuration space is sampled randomly according to various sampling strategies. Samples in the free space, normally called *milestones*, are kept; the samples in collision with obstacles are usually discarded. Nearby pairs of milestones are then connected in a straightforward manner. For example, if we know for every milestone its configuration space distance to an obstacle, we can use this information to argue that a straight line path (in configuration space) between two nearby milestones is fully in free space. These local connections transform the milestones into an undirected graph that constitutes the probabilistic roadmap for the problem. Now given an initial and a final configuration of the robot, we can first connect them to the roadmap and then search this graph for a path between the two configurations. Variations of this basic idea have been demonstrated to work on problems that were infeasible by any other means. The simplicity of this randomized method and its early successes have made it a favorite subject of study in the motion planning community in recent years. A number of different sampling strategies have been developed, including shrinking the obstacles [HKL+98], sampling near the free space boundaries [ABD+98] or medial axis of the configuration space [WAS99], and using "guards" to reject unwanted samples. Industrial adoption has been reported as well.

Nevertheless, a PRM planner does not always succeed to find a path. A typical case arises when a narrow passage in the configuration space connects two large regions where the initial and final configuration lie — this small but critical passage may not be adequately sampled by a PRM strategy. Various refined sampling strategies have been proposed to address this problem [HKL+98, ABD+98, BOvdS99]; in the end, none of them can guarantee that a path will be found when a path exists. Thus, when a PRM planner fails, we never know whether there is actually no path between our two configurations, or whether we simply failed to generate a set of milestones sufficient for discovering the path. This leads us to the topic of this paper, which is *disconnection proofs*, i.e., proofs showing that no path is possible between two robot configurations in a given environment.

Such proofs can be used to terminate early an unending search for a non-existing path by a PRM planner and settle the motion planning problem. In terms of a configuration space view, however, computing disconnection proofs looks like a formidable task. While a randomized planner needs only to be lucky enough to find a path among the many possible ones, a disconnection proof must establish that no such paths exists at all. One may imagine establishing disconnection by randomly sampling the region occupied by obstacles and then constructing a topological sphere that lies fully inside the obstacle region and separates the initial and goal configurations. However, even when the the number of DoFs k (i.e. the dimension of the configuration space) is modest, the surface that we seek is a $k-1$ manifold in a k-dimensional space — and therefore something a lot more complex than the 1-dimensional path that a path that a PRM planner has to find. Surface reconstruction from scattered data is still an active research even for $k = 3$ [EM94, ABK98, CL96]. Thus another approach is needed.

The insight behind our work is to make formal the notion that some objects are 'too big' or 'too long' to be moved from one location to another, as in the informal use 'you cannot take the desk out of the office because it is too big to fit through the door.' More concretely, we can establish that one object does not fit inside another simply by demonstrating that the volume, or the diameter (or one of several other possible functionals on the objects) of the containee is larger than that of the container. Thus we look for disconnection proofs of a special nature, proofs which are compact and not so hard to find. The search for a special proof will typically involve the solution of an optimization problem or a random sampling strategy over a certain space of functionals. Just as a PRM planner might run and not find a path, our disconnection prover might run and not find a disconnection proof. What we aim to guarantee is that if the motion planning problem is 'obviously impossible' within our class of available proofs, then we will find such a proof quickly. Even when our prover fails, its failure can suggest milestones that are of use to a PRM planner. Thus we view a PRM planner and a disconnection prover as working hand-in-hand trying to attack the motion planning problem from both ends. Except in very borderline feasibility cases, we can then hope that one of them will succeed in definitely settling the path existence question.

We report in this paper on some preliminary results along these directions.

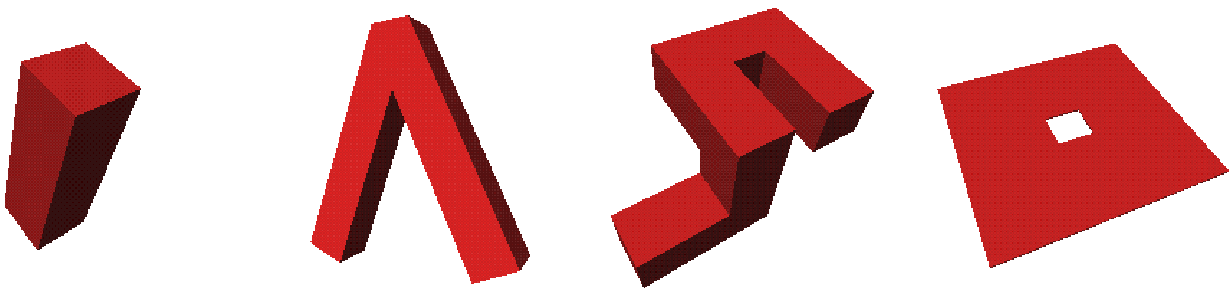
2 Passing through the gate

In order to make the above ideas concrete, we confine ourselves to the following limited, but still quite interesting setup. The *robot* R is a polyhedron moving rigidly in E^3. The environment is free of obstacles except for a *gate* G, which is a polygonal hole in an otherwise solid and impenetrable wall modeled as a flat plane. The motion planning problem is to find a rigid motion of the robot that takes it from a placement in one halfspace defined by the wall, through the gate, to the other halfspace. This is a type of setup in which PRM planners have often been demonstrated. The goal of the disconnection prover is to show that such a motion does not exist.

We start with the following observation. Let α be a constant, $0 < \alpha < 1$ (we will typically choose $\alpha = 1/2$). As the robot goes through the gate, by continuity, there will have to come a moment when exactly a fraction α of the volume of the robot remains in the halfspace of the original robot pose (and therefore $(1-\alpha)$ of the volume is in the halfspace of the destination pose). Let d denote the orientation of the robot at that moment (we think of d as a point in the sphere of directions), and let S_d denote the *section* of the robot by the obstacle plane. Clearly S_d is a polygon, and S_d 'fits' in the gate G. The nature of our disconnection proof will be to establish that, for all directions d, the section S_d of R in orientation d partitioning the volume of R is the ratio $\alpha : 1-\alpha$, cannot fit in the gate G. The latter is an assertion about polygons in the plane: we assert that there is no rigid 2D transformation that can place S_d so that it is inside G. For a given direction d, we call the plane that partitions R in two pieces with ratio $\alpha : 1-\alpha$ the *alpha-plane along* d. The intersection of the robot with the α-plane along d is called the α-section of R along d.

We see immediately some of the difficulties that our approach has to face: as mentioned above, while a PRM planner need only return a path whose non-interference with the obstacles we can check, a disconnection prover has to show something about all possible paths the robot might take. In the above approach we have to consider *all* orientations d for the robot, and for each such orientation we have to consider *all* placements of the corresponding α-section S_d in the gate G. What comes to our rescue is the notion of proof continuity, a concept heavily exploited in the kinetic data structures framework of Basch, Guibas and Hershberger [BGH97]. For example, suppose we have a proof that for a certain specific direction d, the section S_d does not fit in G. Then the same proof is likely to work for a neighborhood of directions around d. Suppose we can find a small θ, even by a very conservative approximation, so that, for every other direction d' within a disk of radius θ around d, the same proof works. Then a complete proof can be constructed by sampling the sphere of directions sufficiently densely so that the set of these disks around the sampled directions fully covers the sphere. A similar idea can be used to replace the infinite set of placements of a given S_d against G by a finite, and therefore computable, set.

We find these proof neighborhoods as follows. Given the robot R, we compute a 'shrunken' version R' of R by an amount ε. The shrinking factor is such that if the α-section of R' in direction d fits in the gate, then the α-section of R along any d' close to d also fits in the gate. Note that R' is not a scaled version of R. It is obtained by removing a layer of thickness ε from R — this is also known as the Minkowski difference of R with a ball of radius ε.



Box *Lambda* *Snake* *Gate*

Figure 1: Robots and a gate.

The main algorithmic components of our disconnection prover will therefore be as follows: (1) computing the α-section of the robot along a given direction, (2) shrinking a polyhedron by an amount ε, (3) testing the inclusion of one polygon in another, and (4) covering the sphere of directions with disks. We will describe each of the above steps in more detail in the following sections. To be specific, we will consider three possible robots, the *box*, the *Lambda*, and the *Snake*; see Figure 1 . The gate is always a simple rectangle, also shown in Figure 1. The robot shapes are constructed in a parametrized manner, so that for certain values of the parameters the robots fit through the gates and for other values not. We will demonstrate the results of running our disconnection prover on these three robots. We will also show how failure of the prover generates milestones that can help and PRM planner.

3 Sectioning a polyhedron at a given volume fraction

In this section, we discuss how to find a cutting plane with a given normal direction d that partitions the polyhedral robot R into two (not necessarily connected) pieces in volume ratio $\alpha : 1-\alpha$; such a plane is clearly unique. Without loss of generality, we assume that the faces of the robot are triangulated.

Note that the volume of a given polyhedron can be computed easily as follows. First, we select an arbitrary reference point, say the origin. The volume of the polyhedron can be computed by summing up all the signed volumes of pyramids having the reference point as a vertex and faces of the polyhedron as their bases (the sign of a pyramid depends on whether its base faces toward the origin or away from the origin). Because of our assumption that all polyhedral faces are triangulated, these pyramids are actually just tetrahedra. Similarly, the volume of the polyhedron above a sectioning plane can be computed by summing up the volumes of the individual pyramids above the plane.

Let us assume that d is the direction vertically up, so we are seeking a horizontal halving section of the robot. If we imagine a horizontal sectioning plane translating upwards, the volume of the polyhedron above the plane decreases. Within a translation interval in which no vertices of the polyhedron are encountered, the volume above the plane of each pyramid is a cubic polynomial of the plane's offset from the origin. It follows that the cutting volume is a cubic polynomial for each interval defined by sorting the vertices of the polyhedron according to their z coordinates. The coefficients for this cubic polynomial in each interval can be determined from the volumes obtained with (any) four different offset values within the interval.

Given a desired section volume, we first find the interval containing this section by performing a binary search among the z coordinates of the vertices. We then compute the coefficients of the cubic polynomial for the volume function in that range, and solve a cubic equation to find the correct sectioning offset. If the polyhedron has n vertices, the cost of the computation is $O(n \log n)$ for sorting the vertices according to their z coordinate, plus $O(\log n)$ volume computations, each costing $O(n)$. Thus the total cost using this approach is $O(n \log n)$.

4 Shrinking polyhedra and polygons

As we saw, it is impossible to consider all sections S_d of R and prove, for each of them, that they do not fit in the gate G. We focus instead on a method to obtain a proof of disconnection that is valid for all directions in a neighborhood around a given direction d. Having done that, it is enough to cover the sphere of directions with small disks, each with its own proof of disconnection.

Consider a direction d, and denote by S_d the α-section of R along d. Let $R^\varepsilon = R \ominus S(0, \varepsilon)$, i.e. R^ε is the set of points in R that are at distance at least ε from the boundary of R. Finally, denote by S_d^ε the α-section of R^ϵ.

Lemma 4.1 *Let δ be the diameter of the projection of R on the plane orthogonal to direction d. If S_d^ε does not fit in the gate G, then $S_{d'}$ does not fit in G for all d' such that $\widehat{d,d'} < \frac{\varepsilon}{\delta}$.*

Proof: Let π be the α-plane of R along d, and d' be a direction with $\widehat{d,d'} = \theta \leq \frac{\varepsilon}{\delta}$. Let π' be the α-plane of R along d'. Consider the rotation ρ whose axis ℓ is the intersection of π and π', and whose angle is θ, such that $\rho(\pi) = \pi'$.

Denote by P the projection of R on the plane orthogonal to d. Note that ℓ passes through the convex hull of P. Indeed, if this was not the case, the fraction of R behind π and π' could not remain the same. Hence, the distance between any point of R and ℓ is at most δ.

Now, consider a point of $p \in R_\varepsilon \cap \pi = S_d^\varepsilon$. Its image $\rho(p)$ is at distance at most $\delta\theta \leq \varepsilon$ from p, and it is therefore contained in R, hence in $S_{d'}$. ◇

In order to remain in the polyhedral domain, we actually compute the Minkowski difference of R with an ε-sided cube rather than a sphere — methods for this are given in [BGRR96, KO92]. An examination of the above proof shows that we could further refine the method by taking the Minkowski difference of R with a box of side length $\theta\delta$ along d and $\theta^2\delta$ along two other orthogonal directions. In our current implementation, due to time constraints, we only compute a conservative approximation to this Minkowski difference, using specialized code for each of our simple robot shapes.

We are now reduced to the following problem: given two (not necessarily simple) polygons S and G, does there exist a rigid motion of P that makes it fit inside G. This problem, being of dimension three (two translational and one rotational DoF), could be solved by exact methods decomposing the three-dimensional configuration space. In our current implementation, however, we take the same approach as for 3D rotations: we cover the circle of 2D directions with small arcs. For each arc, we shrink S so as to guarantee that, if the shrunken version of S cannot be translated so as to fit into G, then S cannot fit into G for any direction in the arc considered. Moreover, as our gate G is convex, we take the convex hull of S before shrinking.

5 Proving the non-containment of polygons

The polygon containment problem has been an active area of research in computational geometry with a long history. There are many variants, according to the type (convex or not) and number of polygons involved, as well as according to the transformations allowed (translation, rotation, scaling); a survey of this literature will be given in the full paper. We are primarily interested in the question of whether a polygon P (the shrunken section S_d, which may actually be a collection of several polygons) fits under translation within the gate G, which we assume to be a simple polygon. The earlier discretization of the circle of directions and the shrinking of the section take care of the rotational aspect of the problem.

If P and Q are polygons with m and n vertices respectively, the translational containment problem can be solved by computing the embedding of the convolution [GRS83] of P and Q in the plane, which in the worst case can take time $O(m^2n^2)$; see in addition [AB87]. It is also known that this problem is 3SUM-hard, and thus likely to be at least $\Omega(mn)$. For our purposes it is not clear that these rather theoretical methods are the most appropriate — because again they essentially construct an arrangement representing the full configuration space of possible placements.

For the instances we have implemented, though, the gate G is always a convex polygon, and this simplifies the problem significantly. Without loss of generality we can then also replace S_d, with its convex hull, from now on abbreviated as S. Thus we have a convex polygon containment problem under translation, which is much simpler, because the configuration space region corresponding to placements of S in G has only size $O(n)$ (the size of G) and it can be computed in time $O(m+n)$; the straightforward details are akin to a standard Minkowski sum computation and are omitted. It is worth noting that Helly's theorem additionally implies that if S cannot be placed into G, then a contradiction can always be obtained from placing only three of the sides of G in the Minkowski sum (convolution, to be more exact): the intersection of their corresponding halfspaces will be empty. Thus there is always a non-containment proof whose size is $O(1)$.

We have not yet implemented a general simple polygon non-containment checker. If the gate has only a small number of reflex vertices or, equivalently, it can be decomposed into a disjoint union of a small number of convex polygons, then we expect that the Helly-type approach mentioned above will work well. Other types of non-containment proofs can be found by considering the largest homothet of a convex shape (say a segment, or a disk) that can fit in S and G. If the size that fits in G is less than that in S, we have non-containment. More generally, we can consider a class of non-negative functionals f for which the problem of computing

$$\mu = \min \int_{\tau(S)} f$$

over all translations $\tau(S)$ of S is not so hard. If there is an f for which $\mu > \int_G f$, we have again shown non-containment. An appropriate f might be found by random sampling, or by a more sophisticated search involving optimization.

6 Sampling the sphere of directions

In certain directions, it is 'obvious' that the robot does not fit through the gate. In others, the robot might barely not make it. (Think of a pencil wanting to go through a small square gate.) To take advantage of this non-uniformity, we cover the sphere of directions adaptively, replacing disks for which it is impossible to prove disconnection by smaller and smaller disks until a proof of disconnection is found or a maximum level of refinement is reached.

More precisely, we start with a rough tessellation of the sphere made of eight congruent spherical triangles, and then process the triangles one after the other. A triangle T is processed as follows: we compute the direction d orthogonal to the plane passing through the three vertices of the triangle, and compute the angle between this d and one of the vertices. This allows us to compute the amount ε by which to shrink the robot so as to prove that all directions within T are blocking. Two cases can happen: if the slice S_d^ε doesn't fit in the gate, we have found a proof and we can discard T. If S_d^ε does fit in the gate, we decompose T into four smaller triangles and process them recursively, until either the previous situation is reached, or we reach a predefined maximum level of refinement.

The 2D prover proceeds in a similar, but simpler, fashion. We cover the circle of directions by four arcs. For each arc, we shrink the polygon and test if the shrunk polygon can be moved by translation into the gate; see Section 5. We subdivide each arc until the shrunken polygon does not fit, or a maximum level of refinement is reached.

In certain cases, our prover will fail because there exists a direction for which the α-slice fits in the gate. We would like to detect such situations to allow for early termination. Whenever S_d^ε fits in the gate, we check if S_d fits in the gate too. If this is the case, the proof of disconnection has failed. We can then return the obtained robot pose, which will constitute a good milestone for a PRM planner (Section 8). Of course it may happen that the prover finds a placement of the shrunken section S^eps_d in the gate, but is unable to verify the same pose for the full robot section S_d. In this failure case no milestone can be returned.

7 Experiments and results

We performed our experiments with the three robot shapes mentioned earlier : the *box*, the *lambda*, and and the *snake*. The gate is always a unit square, and we set $\alpha = 1/2$. Each shape is dependent on one parameter. The parametric shape changes continuously from a form that can easily pass through the gate, to a form that barely fits, to a critical value, to a form that barely does not fit, to a form that clearly does not fit. For instance, in the case of the lambda, we vary the angle between its two legs. In the case of the snake, we vary the length β of its two end links or tines (see Figure 2).

Consider the case of the snake with parameter $\beta = 1.3$. For this shape, our prover was able to find a disconnection proof. Figure 3(a) shows the subdivision of the sphere with which the proof was obtained. As expected, blockage is easier to prove in certain directions than in others. For instance, Figure 3(b) shows the shape oriented in direction $(1, 0, 0)$, with half its volume on each side of the gate. Here, it is obvious that the shape does not fit. Consequently, the triangle around that direction is one of the largest in the sphere subdivision. Compare with Figure 3(c), which shows the snake oriented in direction $(1, 1, 0.3)$ with half its volume on each side of the gate. Here, a very small change of direction completely changes the structure of the section. Hence, very small

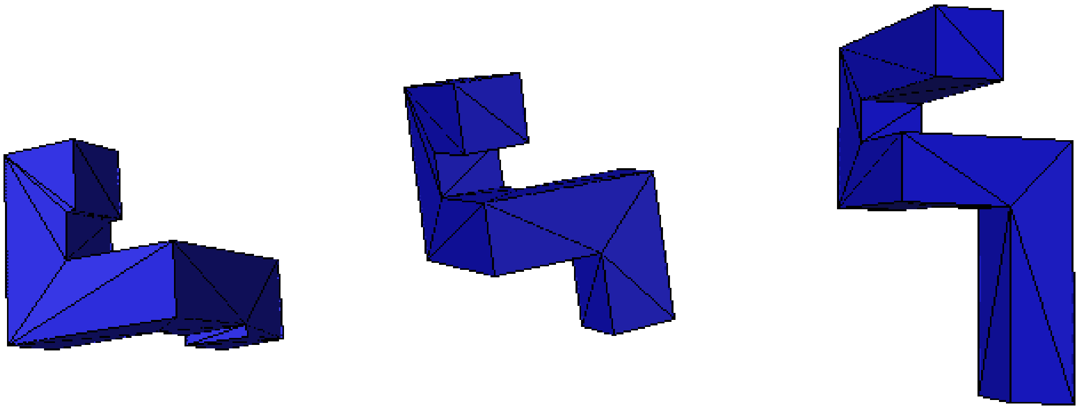


Figure 2: A snake is generated by taking a polygonal skeleton of five links. The three middle links are of length 1.0 and the two end links, or tines, have length equal to the value of the parameter β; the end links shown have lengths 0.4, 1.0, and 2.0 respectively. This skeleton is then swept by a a square of side 0.6 to generate the snake.

triangles were needed in the neighborhood of this direction in order to complete the proof.

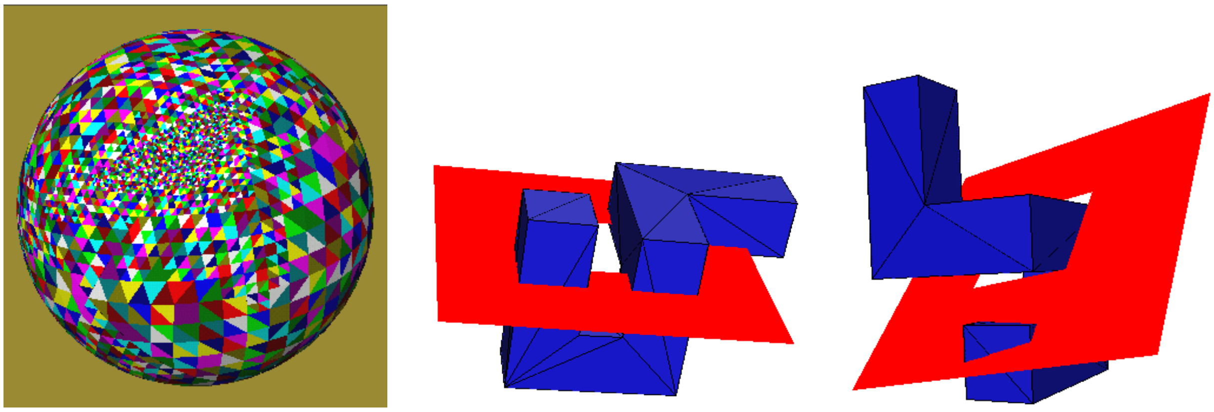
In Figures 4, we report the number of triangles examined by the algorithm before it was able to obtain a disconnection proof or a find fitting placement. In the current implementation on a Linux Intel Pentium 400 MHz, the examination of a single triangle takes roughly 7 ms for the box and 80 ms for the snake. Note that, due to time constraints, no effort has been expended so far at optimizing the code. We show here only the results for the box and for the snake.

In all cases, there exists a critical parameter value above which a disconnection proof can be constructed, and a critical value below which a placement can be found that fits the α-slice in the gate. We will say that the prover fails to decide in the interval between these two values. This failure is due to the fact that we impose a maximum level of refinement of the subdivision. In our experiments, where this level is 8 $(524, 288$ triangles), we actually never experienced a prover decision failure. So in the following, we will assume that the two critical values are very close to each other and talk about a single critical value, which we call β_R.

We see in the graphs that, as the parameter increases towards β_R, the algorithm needs to decompose the sphere of directions into smaller and smaller pieces. This is not surprising, as, close to β_R, a slight shrinking can make a slice fit into the gate even though the original slice does not.

When the parameter goes above β_R, we would expect to have a symmetric behavior: less and less triangles need to be examined as the parameter increases. In our experiments, this is indeed the case for the box shape, but not for the snake. For instance, in the case of the snake, the number of triangles inspected decreases between $\beta = 1.26$ (the first value for which it is possible to find a disconnection proof) and $\beta = 1.6$, but then it starts increasing again.

This is an example of the following paradoxical situation generated by our current approach: in the directions where an object obviously does not pass through the gate, the diameter of its projection is large, and hence the sphere of directions needs to be finely discretized around that direction. The more elongated the object is, the worse the situation. It should be possible to avoid



(a) (b) (c)

Figure 3: (a) The subdivision of the sphere for which a disconnection proof was obtained when $\beta = 1.3$; (b) a direction covered by a large triangle; (c) a direction covered by a small triangle.

this problem if we had a better handle on the locus of rotation axes as defined in the proof of Lemma 4.1. In this case, we should be able to intersect the object with a sphere centered around this locus so as to reduce the diameter of the object considered. This would automatically allow, for a fixed ε, to increase the size of a triangle for which a common proof is can be obtained. There is a trade-off here that requires further careful examination.

Let us say that β_0 is the true value of the parameter below which there exists a path for our robot to pass through the gate. How far is the critical value β_R from β_0? In the case of the box, the two values are clearly equal, but this a rather simple motion planning example. In the case of the snake, with simple calculations, we can calculate that β_0 is between 0.7 and 1.03, which is rather far from $\beta_R \approx 1.25$. This is the price we have to pay to obtain a proof of disconnection that remains tractable. Once again, the goal is not to be complete, but to avoid trying to find a path when one 'obviously' does not exist.

Many extensions and variations of our scheme are clearly possible. If the disconnection fails for a particular value of α, other values can be tried. An extension to the case where the obstacle wall and the gate have some finite thickness is straightforward and will be described in the full paper. In a more general obstacle environment, the gate situation can be simulated by selecting cutting planes through which the robot must pass and in the which the free space is a limited as possible.

8 Integrating the prover with a PRM planner

PRM planners are randomized algorithms with single-sided error. If a PRM planner returns with a collision-free path, it is always a correct solution. However, if it reports that no path exists, a collision-free path connecting the initial and the goal configuration may or may not actually exist. In fact, if no path exists, PRM planners do not even have an effective way of terminating. Most of them simply pre-select a maximum number of milestones to be sampled and reports that no path exists, if that number is reached, and no path has been found. Although it might be possible to estimate the probability whether a path exists based on the milestones already sampled, we can

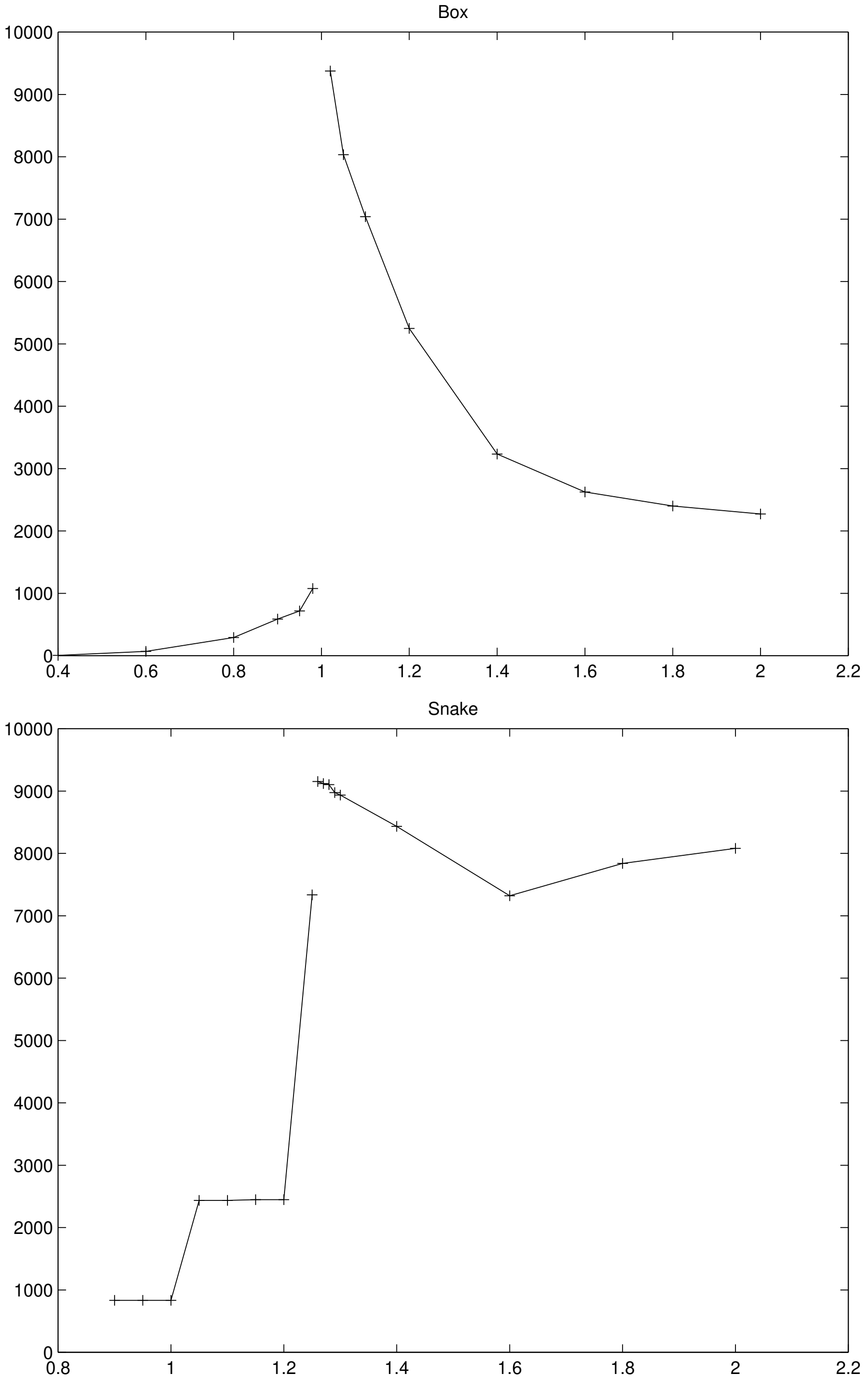


Figure 4: Number of triangles examined on the sphere of directions.

never be sure whether indeed no path exists via sampling alone. On the contrary, the disconnection prover attacks the problem from the other end. When it reports that no path exists, we know for sure that it is a correct answer, but if it fails to find such a proof, then a path may or may not actually exist. Thus the functions of a PRM planner and a disconnection prover are complementary. We can run both of them in parallel. If the prover reports that no path exists, then we can terminate the planner with no uncertainty. Similarly, if the planner returns a collision-free path, there is no need to run the prover any more.

In addition, if for a certain orientation d, we fail to find a non-containment proof of the section S_d in the gate G, the algorithm of Section6 will, in most cases, generate a valid placement of S_d as a by-product. This placement corresponds to a configuration q of the robot partially through the gate, and can be used as a milestone in a PRM planner. Such a configuration q is particularly interesting if a slight perturbation of q causes the robot to collide with obstacles, because it suggests that q lies in a narrow portion of the free space, which may be critical for finding a path. So we can pass these milestones on to the PRM planner and sample more densely in the neighborhood of these milestones. To get more milestones this way, we can also choose different volume ratios to section a polyhedron while running the prover. In this sense, we can view the disconnection prover as a sampler for obtaining interesting milestones.

Work on integrating the prover with our PRM planner is currently underway. We expect to report experimental results on the integration in the final version of the paper.

9 Conclusions

We have demonstrated how a disconnection prover works in a simple setting. Though somewhat special, the examples above fit well the framework of moving pipes and other mechanical assemblies through cluttered spaces and may not be so far from practical utility. There is a very rich area to explore about the kinds of disconnection proofs that can used in different environments. It is also of interest to study conditions on the environment and the robot under which one can prove that a disconnection prover will be guaranteed to find a disconnection proof, if one exists. But in practice, just like a PRM planner may not always find a path when one exists, a disconnection prover may not always find a proof of disconnection, even though no path actually exists.

When a disconnection prover succeeds, the result can be useful in a variety of ways. We believe that most disconnection provers, like the one shown here, will be 'information lossy,' in that they get a contradiction only from a small set of features in the environment or the robot. As such, their disconnection proofs will typically apply not only to the given motion planning problem but to an entire neighborhood of motion planning problems around the given one. In our gate example, an impossibility proof disposes of all problems where the robot has to go from one side of the wall to the other. Furthermore, since we favor simple disconnection proofs, the reason for the disconnection can be useful feedback to a designer on how to modify the robot, or even the environment, so as to make the motion planning task more likely to succeed.

References

[AB87] F. Avnaim and J.-D. Boissonnat. Simultaneous containment of several polygons. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 242–250, 1987.

[ABD+98] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pages 155–168, 1998.

[ABK98] Nina Amenta, Marshall Bern, and Manolis Kamvysselis. A new voronoi-based surface reconstruction algorithm. *Proceedings of SIGGRAPH 98*, pages 415–422, July 1998. ISBN 0-89791-999-8. Held in Orlando, Florida.

[BGH97] J. Basch, L. J. Guibas, and J. Hershberger. Data structures for mobile data. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 747–756, 1997.

[BGRR96] J. Basch, L.J. Guibas, G.D. Ramkumar, and L. Ramshaw. Polyhedral tracings and their convolution. In *Proc. 2nd Workshop on Algorithmic Foundations of Robotics*, 1996.

[BOvdS99] V. Boor, M. H. Overmars, and F. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 1999.

[Can88] J. F. Canny. *The Complexity of Robot Motion Planning.* MIT Press, Cambridge, MA, 1988.

[CL96] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. *Proceedings of SIGGRAPH 96*, pages 303–312, August 1996. ISBN 0-201-94800-1. Held in New Orleans, Louisiana.

[EM94] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13(1):43–72, January 1994.

[GRS83] Leonidas J. Guibas, L. Ramshaw, and J. Stolfi. A kinetic framework for computational geometry. In *Proc. IEEE Symp. on Foundations of Computer Science*, pages 100–111, 1983.

[HKL+98] D. Hsu, L. Kavraki, J. C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In P. K. Agarwal et al., editors, *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pages 141–154. A. K. Peters, Wellesley, MA, 1998.

[KO92] A. Kaul and M. A. O'Connor. Computing minkowski sums of regular polyhedra. Report RC 18891 (82557) 5/12/93, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, 1992.

[KŠLO96] Lydia Kavraki, Petr Švestka, J. C. Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Trans. on Robotics and Automation*, 12(4):566–580, 1996.

[Lat91] J. C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991.

[WAS99] S A. Wilmarth, N. M. Amato, and P. F. Stiller. Maprm: A probabilistic roadmap planner with sampling on the medial axis of the free spac. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1024–1031, 1999.