Star-shaped Roadmaps - A Deterministic Sampling Approach for Complete Motion Planning

Gokul Varadhan Dinesh Manocha
University of North Carolina at Chapel Hill
http://gamma.cs.unc.edu/motion/
Email: {varadhan,dm}@cs.unc.edu

***Abstract*— We present a simple algorithm for complete motion planning using deterministic sampling. Our approach relies on computing a *star-shaped* roadmap of the free space. We partition the free space into star-shaped regions such that a single point called the guard can see every point in the star-shaped region. The resulting set of guards capture the intra-region connectivity. We capture the inter-region connectivity by computing connectors that link guards of adjacent regions. We use the guards and connectors to construct a star-shaped roadmap of the free space. We present an efficient algorithm to compute the roadmap in a deterministic manner without computing an explicit representation of the free space. We show that the star-shaped roadmap captures the connectivity of the free space while providing sufficient information to perform complete motion planning. Our approach is relatively simple to implement for robots with translational and rotational degrees of freedom (dof). We highlight the performance of our algorithm on challenging scenarios with narrow passages or when there is no collision-free path for low-dof robots.**

I. Introduction

Motion planning is a fundamental problem in robotics and has been extensively researched for more than three decades. We address the problem of planning the path of a robot navigating through a static environment. At a broad level, prior motion planning algorithms can be classified into exact criticality-based algorithms and approximate approaches [7]. Some of the early work on criticality-based algorithms includes exact free-space computation, roadmap methods, and exact cell decomposition methods. These approaches perform *complete* planning – they find a collision-free path if one exists, or guarantee that there is no collision-free path from the initial to the goal configuration. However, these algorithms have a high theoretical complexity and are difficult to implement in practice for general robots. As a result, most practical algorithms for complete planning have been restricted to rigid planar objects, 3D convex polytopes or special objects (e.g. ladders, discs or spheres). Given the underlying complexity of exact motion planning, a number of approximate approaches have been have been proposed. These include approximate cell decomposition, potential-field methods and randomized sampling based methods. The approximate cell decomposition methods can be made *resolution complete*, provided the resolution parameters are chosen appropriately. Many of the current planners compute a probabilistic roadmap using techniques based on randomized sampling [5]. These methods are simple to implement and have been successfully applied to high-dof motion planning problems in different applications. However, the approximate algorithms may not guarantee completeness, especially when there is no collision-free path.

Main Results: We present a new motion planning algorithm for robots with translational and rotational dof. Our work combines the simplicity of sampling-based approaches with the completeness of exact algorithms. We compute a sampling of the free space in a deterministic manner using an adaptive volumetric grid. We generate sufficient number of samples to capture the connectivity of the free space, as long as there is no tangential contact on the boundary of the free space. As a result, we are guaranteed to find a collision-free path if one exists or detect non-existence of any collision-free path.

Our approach is based on the notion of *star-shapedness*. A region R is *star-shaped* if there exists a point $\boldsymbol{o} \in R$, called a *guard*, that can see every point $\boldsymbol{p}$ in the region, i.e., the straight line segment $\boldsymbol{op}$ does not intersect the boundary of R. We show that star-shapedness provides a compact *encoding* of the connectivity of a region. We decompose the free space into star-shaped regions without computing an explicit representation of the free space. The resulting set of guards capture the intra-region connectivity for each region. Furthermore, we capture the inter-region connectivity by computing *connectors* that connect guards of adjacent regions. We use these guards and connectors to construct a *star-shaped* roadmap of the free space.

The underlying computation in our planner is the star-shaped test. We present a simple and efficient algorithm that uses linear programming and interval arithmetic to perform this test. Unlike prior criticality-based methods, we are able to avoid exact computation of roots of algebraic equations and are able to perform conservative star-shaped tests for early termination. As a result, our algorithm is relatively simple to implement. In the worst case, the complexity of our algorithm can increase exponentially with the number of dof. We also compare some features of our approach with approximate cell-based decomposition and randomized sampling based algorithms. We have implemented our planner and demonstrated its performance to compute collision free paths for low dof robots in challenging scenarios: when there are narrow-passages in free space or no collision-free paths.

Organization: The rest of the paper is organized in the following manner. We give an overview of related work in motion planning in Section II. In Section III, we give a brief overview of configuration space formulation and present the notation used in the rest of the paper. We present star-shaped roadmaps in Section IV. We present our deterministic sam-

pling algorithm in Section V and describe its implementation in Section VI. We compare our approach with prior approaches in Section VII and discuss a few limitations.

II. Previous Work

Motion planning has been extensively studied in the literature for more than three decades. A comprehensive survey of motion planning results is presented in [6], [7].

A. Exact Approaches

There are two main approaches for exact or complete motion planning. These approaches are based on roadmap computation and cell decomposition. Examples of a roadmap-based approach include the visibility graph method, retraction approach [6], and the silhouette method [3]. Exact cell decomposition methods have been extensively studied for motion planning and the first complete algorithm was proposed by Schwartz and Sharir [10]. The details of the above methods are quite involved and are not easy to implement. A number of complete algorithms have been proposed for restricted cases of motion planning problem – including rigid planar objects with 3dof, 3D convex polytopes, 3D polyhedral objects with only translational dof, and special objects in 3D such as ladders, discs, or balls [6].

B. Approximate Cell Decomposition and Sampling Based Approaches

A number of algorithms based on approximate cell decomposition have also been proposed [8]. These methods partition the configuration space into a collection of cells. They classify the cells into three types: *empty* cells that lie completely in free space, *full* cells that are completely within C-obstacle, and *mixed* cells that contain the boundary of the free space. The set of empty cells provide a conservative approximation of the free space and are used for path computation. The approximate cell decomposition methods are *resolution complete*, i.e., they can find a path if one exists provided the resolution parameters are selected small enough [6]. They have been used for low dof robots.

The probabilistic roadmap method (PRM) [5] is perhaps the most widely used path planning algorithm for different applications. It is relatively simple to implement and has been successfully used for motion planning of high dof robots. Since PRM-based algorithms sample the free space randomly, they may fail to find paths – especially those passing through narrow passages. A number of extensions have been proposed to improve the sampling in terms of handling narrow passages [1], [16] or using visibility-based techniques [11]. All these methods are *probabilistically complete*. Some extensions of PRMs have been proposed that may be able to detect non-existence of a path [2].

III. Preliminaries

A. Configuration Space and Contact Surfaces

We assume that the robot $\mathcal{A}$ is a rigid or an articulated object moving among stationary rigid obstacles $\mathcal{B}$. We also assume that the geometry of both $\mathcal{A}$ and $\mathcal{B}$ is accurately known. The free space $\mathcal{F}$ is the set of configurations at which $\mathcal{A}$ does not collide with $\mathcal{B}$. The boundary of $\mathcal{F}$, denoted as $\partial\mathcal{F}$, consists of those configurations of $\mathcal{A}$ at which $\mathcal{A}$ makes contact with $\mathcal{B}$, but does not penetrate into the interior of $\mathcal{B}$. Therefore, $\partial\mathcal{F}$ can be expressed in terms of a collection of *contact surfaces* (C-surfaces), each being the locus of configurations of $\mathcal{A}$ at which a specific feature of $\mathcal{A}$ is in contact with a feature of $\mathcal{B}$. We refer the reader to [6] for a detailed explanation of the configuration space formulation and C-surfaces.

We use two important properties of C-surfaces for generating star-shaped roadmaps:

Superset property: The set Γ of C-surfaces is a superset of the boundary $\partial\mathcal{F}$ of free space, i.e., $\partial\mathcal{F} \subseteq \bigcup\{\gamma_i \in \Gamma\}$. Γ defines an arrangement and $\mathcal{F}$ is a subset of the cells in this arrangement. Each cell defines one connected component of $\mathcal{F}$.

Orientation property: We can assign an orientation to each C-surface. We explain this with an intuitive argument. Consider a C-surface γ generated by the contact between a robot feature f_1 and an obstacle feature f_2. Points on *one side of* γ correspond to the case where f_1 has penetrated f_2 and points on the *other side* correspond to no overlap or contact between f_1 and f_2. We orient γ by assigning a normal at $\boldsymbol{p}$ to point in the direction of no overlap.

B. Notation

We use the following notation in the rest of the paper. We use lower case bold letters such as $\boldsymbol{p}, \boldsymbol{q}$ to refer to points in $\mathbb{R}^d$. We use the symbol $\boldsymbol{pq}$ to refer to the line segment between the points $\boldsymbol{p}$ and $\boldsymbol{q}$.

$\mathcal{C}$ denotes the configuration space. $\mathcal{F}$ denotes the free space and $\partial\mathcal{F}$ denotes its boundary. The letter $R \subseteq \mathcal{C}$ denotes a region in the configuration space. Γ denotes the set of C-surfaces that contribute to the boundary of the C-obstacle.

A restriction of a set S w.r.t another set T is denoted as S_T and is defined $S \cap T$. We assume S_T is a closed set.

A surface γ defined in $\mathbb{R}^d$ is *star-shaped* if there exists a point $\boldsymbol{o} \in \mathbb{R}^d$ (called the origin) such that $\boldsymbol{op} \cap \gamma = \{\boldsymbol{p}\}\ \forall\ \boldsymbol{p} \in \gamma$. Given a star-shaped region R, let $R^\star = \boldsymbol{o}$. Similarly, if a point $\boldsymbol{p} \in R$, then let $\boldsymbol{p}^\star = \boldsymbol{o}$

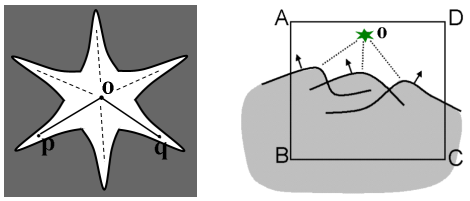
Given a set S, two points $\boldsymbol{p}, \boldsymbol{q} \in S$ are connected if there exists a path between $\boldsymbol{p}$ and $\boldsymbol{q}$ that lies in S. We use the shorthand notation $\boldsymbol{p} \overset{S}{\longleftrightarrow} \boldsymbol{q}$ to mean $\boldsymbol{p}$ and $\boldsymbol{q}$ are connected in S. The connectivity relation is symmetric. Given a roadmap (an undirected graph) $\mathcal{R} = (V, E)$ and two vertices $\boldsymbol{v}, \boldsymbol{w} \in V$, $\boldsymbol{v} \overset{\mathcal{R}}{\longleftrightarrow} \boldsymbol{w}$ means that $\boldsymbol{v}$ and $\boldsymbol{w}$ are connected in $\mathcal{R}$, i.e., there exists a path between $\boldsymbol{v}$ and $\boldsymbol{w}$ consisting of a sequence of edges in E.

IV. Star-shaped Roadmaps

In this section, we present the concept of a star-shaped roadmap and show that it captures the connectivity of the free space for complete motion planning. We use these properties of star-shaped roadmaps to design a deterministic sampling algorithm in Sec. V.

A. Star-shapedness

Our approach is based on the notion of star-shapedness. A region R is *star-shaped* if there exists a point $\boldsymbol{o} \in R$, an *origin*, that can *see* every point $\boldsymbol{p}$ in the region, i.e., the straight line segment $\boldsymbol{op}$ does not intersect the boundary of R. The origin is commonly referred to as a *guard*. It is easy to show that a star-shaped region is always connected. Moreover, every point in the region is connected to the guard along a straight line segment. Star-shapedness is thus a compact way of encoding



(a) Star-shapedness (b) Star-shaped Test

Fig. 1. *Left : This figure shows a star-shaped region (in white). It contains a guard $\boldsymbol{o}$ that can see every point within the region. A path between any two points $\boldsymbol{p} \in R$ and $\boldsymbol{q} \in R$ is given by $\boldsymbol{po}$ :: $\boldsymbol{oq}$. Right: If in a region R, all C-surfaces are star-shaped w.r.t a common point $\mathbf{o}$, then $\mathcal{F} \cap R$ is star-shaped w.r.t $\mathbf{o}$.*

the connectivity of a region. It provides a path between every point in the region and the guard. We exploit this property for motion planning. A path between any two points $\boldsymbol{p} \in R$ and $\boldsymbol{q} \in R$ is given by $\boldsymbol{po}$ :: $\boldsymbol{oq}$ where :: denotes path concatenation (see Fig. 1(a)). We extend this idea to compute a path between two arbitrary configurations in free space.

B. Overall Approach

At a conceptual level, our approach computes a *star-shaped decomposition* of the free space, i.e., it partitions $\mathcal{F}$ into a set of star-shaped regions. We present our algorithm to compute the star-shaped decomposition in Sec. V. Based on star-shapedness, we capture the intra-region connectivity. However, we also need to take into account the inter-region connectivity, i.e. connectivity between points belonging to separate regions. We achieve this by computing *connectors* [1]. Our approach consists of the following steps:

1) Compute a star-shaped decomposition Σ of the free space into star-shaped regions R.
2) For every pair of adjacent regions (R_i,R_j) in Σ, compute a point $\boldsymbol{c}$ on the common boundary shared by R_i and R_j. We refer to $\boldsymbol{c}$ as a *connector* – it connects the guards of R_i and R_j.
3) Construct a star-shaped roadmap $\mathcal{R}$ using the guards and connectors computed in Steps 1 and 2.

We illustrate these steps in Fig. 2.

C. Star-shaped Decomposition and Guard Computation

Step 1 computes a star-shaped decomposition of the free space. The resulting set of guards constitutes a sampling of the free space and we refer to it as a *star-shaped* sampling of the free space. The star-shaped sampling provides an implicit description of the free space.

$$\boldsymbol{p} \in \mathcal{F} \iff \boldsymbol{p} \text{ is visible to at least one of the guards.}$$

The concept of star-shaped decomposition is related to the famous art gallery problem [9]. The art gallery problem is concerned with finding the minimum number of guards that can cover a region. In our context, computing a minimum number of guards would be desirable, but not necessary.

[1]We borrow the terms *guard* and *connector* from [11] because these are similar concepts. However, our definitions are different from the ones used in [11].

D. Connector Computation

In Step 2, we capture the inter-region connectivity. It suffices to only consider paths between adjacent regions R_i and R_j that cross their common boundary R_{ij}. We compute a point $\boldsymbol{c}$ belonging to R_{ij}. $\boldsymbol{c}$ is a connector. Since the regions R_i and R_j are star-shaped, $\boldsymbol{c}$ is visible to the guards of R_i and R_j. Hence, $\boldsymbol{c}$ connects the guards of two adjacent regions (see Fig. 2(b)).

E. Roadmap Computation

In Step 3, we combine the guards and connectors to construct a star-shaped roadmap $\mathcal{R}$ of the free space (see Fig. 2(b)). $\mathcal{R}$ is an undirected graph. Let G and C denote the set of guards and connectors. The set of graph vertices is $V = G \cup C$. Each connector $\boldsymbol{c}$ connects two guards $\boldsymbol{g_1} \in G$ and $\boldsymbol{g_2} \in G$ of two adjacent regions. This defines two graph edges $(\boldsymbol{c}, \boldsymbol{g_1})$ and $(\boldsymbol{c}, \boldsymbol{g_2})$. Let $GUARDS(\boldsymbol{c})$ denote the set $\{\boldsymbol{g_1}, \boldsymbol{g_2}\}$. The set of graph edges E is defined as:

$$E = \{(\boldsymbol{c}, \boldsymbol{g}) \mid \boldsymbol{c} \in C,\ \boldsymbol{g} \in GUARDS(\boldsymbol{c})\}$$

$\mathcal{R}$ is the undirected graph (V, E, w) where the weight function $w : E \rightarrow \mathbb{R}$ is defined as a distance between the edge vertices using a suitable metric (e.g. Euclidean).

F. Complete Path Planning

Given the star-shaped decomposition Σ and the roadmap $\mathcal{R}$, path planning becomes straightforward. Let $\boldsymbol{p}$ and $\boldsymbol{q}$ respectively denote the start and goal configuration respectively. Assume they are connected. The star-shapedness property of each region in Σ implies we can connect $\boldsymbol{p}$ and $\boldsymbol{q}$ to the guards $\boldsymbol{p}^\star$ and $\boldsymbol{q}^\star$ respectively by straight line paths. We compute a path between $\boldsymbol{p}^\star$ and $\boldsymbol{q}^\star$ in the roadmap $\mathcal{R}$ based on graph search. The following theorem states that our motion planning algorithm is complete.

THEOREM 1 *A path exists between two points $\boldsymbol{p}$ and $\boldsymbol{q}$ if and only if $\boldsymbol{p}$ and $\boldsymbol{p}^\star$ are connected in $\mathcal{F}$, $\boldsymbol{p}^\star$ and $\boldsymbol{q}^\star$ are connected in $\mathcal{R}$, and $\boldsymbol{q}^\star$ and $\boldsymbol{q}$ are connected in $\mathcal{F}$, i.e.,*

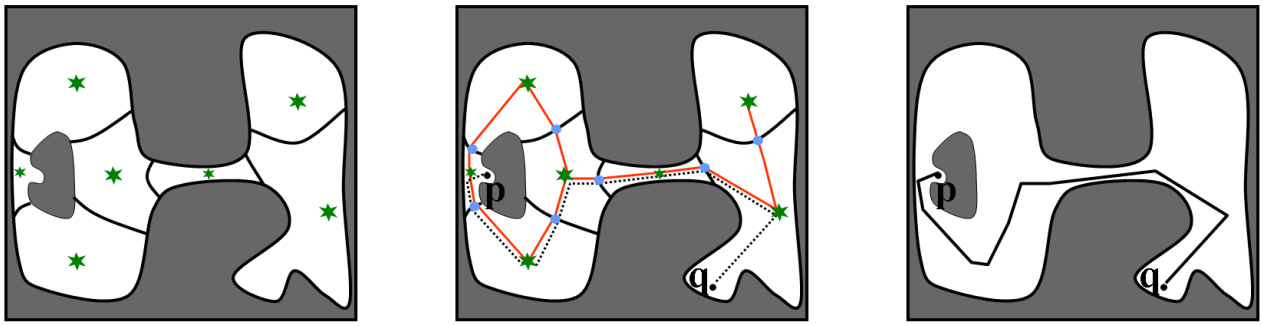
$$\boldsymbol{p} \overset{\mathcal{F}}{\longleftrightarrow} \boldsymbol{q} \iff \begin{array}{l} \boldsymbol{p} \overset{\mathcal{F}}{\longleftrightarrow} \boldsymbol{p}^\star \\ \boldsymbol{p}^\star \overset{\mathcal{R}}{\longleftrightarrow} \boldsymbol{q}^\star \\ \boldsymbol{q}^\star \overset{\mathcal{F}}{\longleftrightarrow} \boldsymbol{q} \end{array}$$

Due to space limitations, we omit the proof. A detailed proof is given in [15].

An important consequence of the above theorem is the following corollary which enables us to find a collision-free path.

COROLLARY 1 Path Planning*: if $\boldsymbol{p} \overset{\mathcal{F}}{\longleftrightarrow} \boldsymbol{q}$, then*

1) *There exists a straight line path α between $\boldsymbol{p}$ and $\boldsymbol{p}^\star$. Similarly, there exists a straight line path β between $\boldsymbol{q}$ and $\boldsymbol{q}^\star$.*
2) *There exists a path δ between $\boldsymbol{p}^\star$ and $\boldsymbol{q}^\star$ in the roadmap $\mathcal{R}$.*
3) *A path between $\boldsymbol{p}$ and $\boldsymbol{q}$ is given by $\alpha :: \delta :: \beta$ where :: denotes path concatenation.*



(a) Star-Shaped Decomposition (b) Star-Shaped Roadmap (c) Path Planning

Fig. 2. *Star-shaped Roadmap: This figure shows how to construct a star-shaped roadmap and its application to path planning. The C-obstacle is shown in gray while the free space is shown in white. We first compute a star-shaped decomposition of the free space (Fig. (a)). Each region in the decomposition contains a guard (green star) that can see every point in the region. We connect guards of adjacent regions by computing connectors (blue circles) on the common boundary between the two regions. The guards and connectors are used to create the star-shaped roadmap as shown in Fig. (b). Fig (c) shows how a path is computed between two points $\boldsymbol{p}$ and $\boldsymbol{q}$ by connecting them to the roadmap and finding a path along the roadmap.*

We use the following corollary of Theorem 1 as a test for non-existence of any collision-free path for complete motion planning.

COROLLARY 2 Path Non-Existence: *If there is no path between $\boldsymbol{p}^\star$ and $\boldsymbol{q}^\star$ in the roadmap $\mathcal{R}$, then there is no collision-free path between $\boldsymbol{p}$ and $\boldsymbol{q}$*

V. A Deterministic Sampling Algorithm

In this section, we present an algorithm to compute a star-shaped roadmap by sampling the free space in a deterministic manner.

A. Configuration Space Subdivision

The approach presented in Sec. IV relied on a star-shaped decomposition of the free space. In practice, we do not have an explicit representation of $\mathcal{F}$ and hence it is not possible to compute such a decomposition explicitly. In fact, an explicit decomposition of the free space is not even required. Instead, we compute a subdivision of the configuration space $\mathcal{C}$ into regions R such that $\mathcal{F}_R = \mathcal{F} \cap R$ is star-shaped. Such a subdivision is sufficient for complete motion planning.

In Sec. V-C, we present a simple algorithm for computing such a subdivision adaptively. Our algorithm relies on the ability to perform two queries:

1) **Free Space Existence query**: Given a region R, determine if R contains a part of free space, i.e. $\mathcal{F}_R \neq \emptyset$.
2) **Star-shaped query**: Given a region R, determine if $\mathcal{F}_R$ is star-shaped.

Our goal is to perform these queries without computing an explicit representation of the free space. Instead of performing exact tests, we present a sufficient condition on R and use it to answer these queries.

B. Star-shaped Test

Consider all the C-surfaces γ_i that intersect R and compute their restriction $\gamma_i \cap R$ to region R. Let Γ_R denote the resulting set of surfaces. We can answer the above queries if R satisfies the following condition:

Star-shaped Test: Are all the surfaces in Γ_R star-shaped w.r.t a common point $\boldsymbol{o}$?

See Fig. 1(b). If R satisfies the above test, then we can answer both the queries. R contains a part of $\mathcal{F}$ if and only if $\boldsymbol{o} \in \mathcal{F}$. Moreover, if this is true, then $\mathcal{F}_R$ is star-shaped w.r.t $\boldsymbol{o}$. Formally, we have the following lemma:

LEMMA 1 *If there exists a point $\boldsymbol{o} \in R$ such that every $\gamma \in \Gamma_R$ is star-shaped w.r.t. $\boldsymbol{o}$, then*

1) **Free Space Existence query**: *$\mathcal{F}_R \neq \emptyset \iff \boldsymbol{o} \in \mathcal{F}$*
2) **Star-shaped query**: *If $\boldsymbol{o} \in \mathcal{F}$ then $\mathcal{F}_R$ is star-shaped w.r.t $\boldsymbol{o}$.*

A detailed proof of the lemma is given in [15].

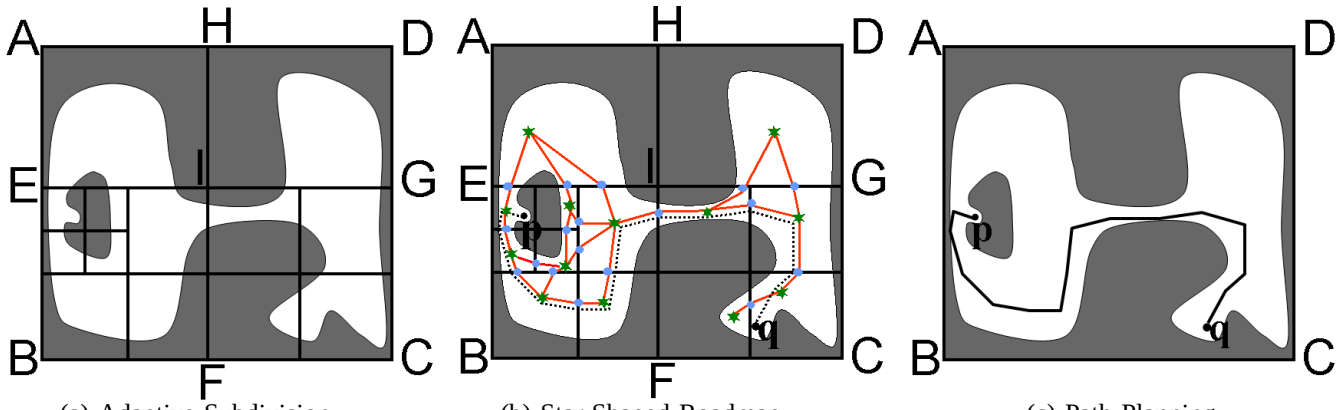
We present a simple technique for performing the star-shaped test in Sec. V-E and to compute Γ_R in Sec. V-F.

C. Adaptive Subdivision and Guard Computation

We generate an adaptive subdivision of $\mathcal{C}$. Our algorithm starts with a region R that bounds the boundary of $\mathcal{F}$. It applies the star-shaped test to R. If this test is satisfied, our algorithm sets the guard for the region R to be the origin with respect to which $\mathcal{F}_R$ is star-shaped. Otherwise, if the star-shaped test fails, the algorithm subdivides R into a set of sub-regions. Then the algorithm is recursively applied to the sub-regions. Fig. 3(a) illustrates the subdivision algorithm in 2D.

D. Connector Computation

The objective of connector computation is to determine if the free space of two adjacent regions, R_i and R_j, are connected through a point on their common boundary R_{ij}. In other words, we wish to test if R_{ij} has a point in $\mathcal{F}$. This problem is almost identical to the free space existence problem discussed in the previous section with one difference – the dimension of R_{ij} is one less than the dimension of the configuration space – hence we can use the same approach presented in the previous section to solve this problem. We subdivide R_{ij} into subregions that satisfy the star-shaped test and then check if any of the origins of any sub-region lies



(a) Adaptive Subdivision (b) Star-Shaped Roadmap (c) Path Planning

Fig. 3. *Star-Shaped Roadmap Construction: This figure shows how we compute a star-shaped roadmap using adaptive subdivision of the configuration space. We subdivide the configuration space into regions R such that the free space contained within R, given by $\mathcal{F} \cap R$, is star-shaped. Fig. (b) shows the star-shaped roadmap that was obtained from the resulting subdivision. Fig. (c) shows how the roadmap is used for path computation.*

in the free space. We merely need to compute just one point in $\mathcal{F}$ (rather than capture all of them). As soon as we find one such point, we stop the subdivision process. This point is classified as a connector. If R_{ij} contains no point in $\mathcal{F}$, then the subdivision process will continue until all the sub-regions satisfy the star-shaped test and none of the corresponding origins lie in the free space. In this case, the free space of R_i and R_j belong to two separate components of the free space.

E. Efficient Implementation of the Star-shaped Test

Given a region R, we need to determine if all the surfaces in Γ_R are star-shaped w.r.t a common point. Here we present an algorithm to test if a single C-surface γ is star-shaped or not. This technique extends directly to the case of multiple C-surfaces. In general, the C-surface is a non-linear algebraic surface defined in a high dimensional configuration space. The exact test is as follows: is there a point $\boldsymbol{o}$ such that $\boldsymbol{n}\cdot(\boldsymbol{o}-\boldsymbol{x}) > 0,\ \boldsymbol{x} \in \gamma$ where $\boldsymbol{n}$ is the normal at point $\boldsymbol{x}$. This test reduces to solving a system of high degree algebraic equations. Instead of computing an exact solution to the algebraic equations, we use a simple and efficient technique to perform the test conservatively.

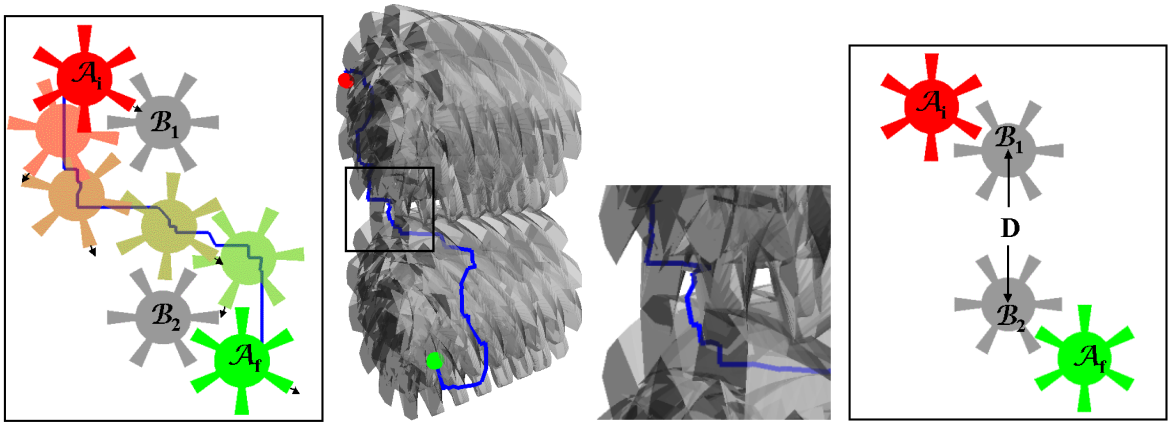
In general, the origin is not a unique point. Given a C-surface γ defined in $\mathbb{R}^d$, there exists a set $Ker(\gamma) \subseteq \mathbb{R}^d$ of points that can be used as an origin. The set $Ker(\gamma)$ is called the kernel of γ. γ is star-shaped if and only if it has a non-empty kernel. The central idea behind our technique is to *guess* a candidate point for the origin and then verify that the candidate point indeed lies within the kernel. Verifying whether a surface is star-shaped w.r.t a fixed point is much easier than performing the exact test. The method proceeds by estimating a candidate point that lies in the interior of an approximate kernel of a sampled version of the surface. The candidate point is computed by linear programming. We then verify if it belongs to the kernel of the surface by interval arithmetic. A detailed explanation of this technique is given in [14]. We provide a summary here.

We take advantage of a parametrization of the C-surfaces [6]. Using the parametric representation, we enumerate a set of points on γ. By the orientation property of C-surfaces (Sec. III), each point $\boldsymbol{x}$ has a well defined normal $\boldsymbol{n}$. The star-shapedness property requires that the origin $\boldsymbol{o}$ should see point $\boldsymbol{x}$; this defines the constraint $\boldsymbol{n} \cdot (\boldsymbol{o} - \boldsymbol{x}) > 0$. In this manner, each point defines a linear constraint on $\boldsymbol{o}$. We use linear programming to check if the resulting constraints admit a feasible solution. The feasible region corresponds to an approximate kernel of the C-surface. If the feasible region is non-empty, then we can choose the center of the feasible region to be the candidate point (see [14]).

If a candidate point $\boldsymbol{o}$ is computed, it is likely to be a valid origin provided we enumerated a sufficient number of points on γ. In general, we do not know how many such points are needed; so we choose a fixed number of points. To ensure correctness, we check for the validity of the resulting candidate point by testing if γ is indeed star-shaped w.r.t $\boldsymbol{o}$. We check if $\boldsymbol{n} \cdot (\boldsymbol{o} - \boldsymbol{x}) > 0\ \forall \boldsymbol{x} \in \gamma$. Since γ is an algebraic surface, it is represented as $f(x, y, z) = 0$. Therefore, the expression reduces to $\bigtriangledown f^T(\mathbf{o} - \mathbf{x}) > 0\ \forall \boldsymbol{x} \in \gamma$. We compute a set of intervals around γ and verify that the above expression is positive within each interval. We use interval arithmetic [12] to perform this test. Using interval arithmetic, we can conservatively check if this property holds for all the points on γ. If a candidate point is computed and verified by the interval arithmetic test, then R is said to have passed the star-shaped test, otherwise R has failed the star-shaped test.

As noted earlier, the test is conservative – a surface γ may be star-shaped, but as per our technique R may fail the test. If R fails the test, we then subdivide R into sub-regions and repeat the test on the sub-regions. Thus, the conservative test may result in some additional subdivisions. However, computation of additional subdivisions does not affect the correctness of the algorithm. This is because if γ is star-shaped w.r.t R, at some level of the subdivision of R, all the subregions of R will satisfy the star-shaped test. This holds because as a sub-region Q shrinks, the approximate kernel of $\gamma \cap Q$ approaches the exact kernel. As a result, the probability that the candidate point lies within the kernel increases as the sub-regions shrink.

In this manner, we are able to use a conservative test and still guarantee complete path planning as long as there are no tangential contacts on the boundary of fee space. Moreover,



(a) Gears 1 (b) Configuration Space (c) Narrow Passage (d) Gears 2 (No path)

Fig. 4. **3-DOF Planning with 2T and 1R**: *This figure highlights application of our algorithm to planar motion planning with both translational as well as rotational dof. Fig. (a) shows a gear-shaped robot navigating amongst two gear-shaped obstacles ($\mathcal{B}_1$ & $\mathcal{B}_2$) (shown in gray). The initial and final locations of the robot are shown in red and green respectively ($\mathcal{A}_i$ & $\mathcal{A}_f$ resp.). The robot is allowed to move only within a bounded workspace (shown as a black rectangle). The figure also shows a number of positions of the robot during its motion along the path. Fig. (b) shows the path in the configuration space (drawn translucently). Fig. (c) shows a zommed view of the narrow passage. Fig. (d) shows a similar environment where the two obstacles are moved closer to each other and as a result, no collision-free path exists.*

we don't use any exact non-linear equation solver. A tangential contact occurs when two C-surfaces touch each other at a point thus forming a narrow passage of width zero in the free space. Our algorithm cannot handle them because the free space in a neighborhood of a tangential contact is never star-shaped – for any arbitrary neighborhood of nonzero volume.

F. Intersection Computation

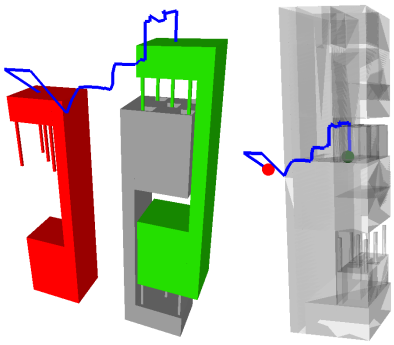
The above approach relies on determining whether a C-surface γ_i intersects a bounded $d-$dimensional region R; if they do intersect, the part of the surface contained within R is computed. We note that R is axis aligned and γ_i has a parametric representation. We determine if γ_i intersects R by performing interval arithmetic. Interval arithmetic is cheap, conservative, and suffices for our purpose.

To obtain $\gamma_i \cap R$, we take advantage of two facts: 1) The C-surfaces γ_i are parameterized in terms of the coordinates of the configuration space and 2) R is an axis aligned region in configuration space. Hence we obtain $\gamma_i \cap R$ by considering a restricted parametric domain $D \cap R$ where D is the entire parametric domain.

VI. Implementation & Results

In this section, we describe the implementation of our algorithm and demonstrate its performance on several motion planning scenarios. We used C++ programming language with the GNU g++ compiler under Linux operating system. Table I reports the performance of our algorithm. All timings are on a 2 GHz Pentium IV PC with a GeForce 4 graphics card and 512MB RAM. Our current implementation is unoptimized.

Fig. 4 highlights application of our algorithm to planar motion planning with both translational as well as rotational degrees of freedom. The robot must pass through a very narrow passage to reach its goal. Moreover, it must undergo both translation as well as rotation. Our algorithm took 111 secs to construct a star-shaped roadmap. Using this roadmap, it took only 0.22 secs to compute a path (shown as a blue curve). Fig. 4 (d) highlights the use of our algorithm to detect nonexistence of a collision-free path. The two obstacles are too close to each other and consequently, the robot cannot pass between them. Our algorithm took 90 secs to compute a roadmap for this environment and detected non-existence of a path in 0.18 secs.



(a) Assembly (b) Configuration Space

Fig. 5. **3D Translational motion planning**: *This example shows application of our algorithm to motion planning of a three-dimensional robot with translational degrees of freedom. It consists of two identical parts each with pegs and holes. The goal is to assemble the two parts so that the pegs of one part fit into the holes of the other. The left image shows a path that the robot can take so that the two parts could be assembled. The right image shows the path in configuration space (drawn transluscently).*

Fig. 5 shows application of our algorithm to a three-dimensional assembly planning scenario. The goal is to assemble two parts such that the pegs of one part fit into the holes of the other. Our algorithm took 16 secs to construct a roadmap and was able to find a path (shown in blue) in 0.22 secs. This is a challenging example because the goal configuration, wherin the pegs fit into the holes, is lodged within a very narrow passage in the configuration space.

Fig. 6 shows application of our algorithm to a **3R** planar articulated robot with 3 revolute joints. Our algorithm took 17 secs to construct a star-shaped roadmap. Using this roadmap, it took only 0.43 secs to compute a path. The robot must pass through a narrow passage to reach its goal. We also experimented with a modified environment where the obstacle is closer to the robot. As a result, no path exists between the initial and goal configurations. Our algorithm took 16 secs to

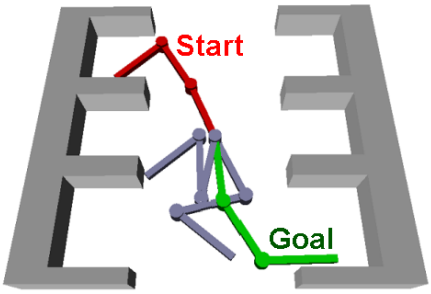


Fig. 6. **3R Articulated Robot**: *This example shows application of our algorithm to motion planning of a planar articulated robot with 3 revolute joints. The figure shows a start configuration, a goal configuration and two intermediate configurations.*

compute a roadmap for the modified environment and detected non-existence of a path in 0.14 secs.

Table I provides the timings of our algorithm on these models. It also provides statistics such as the number of guards and connectors in the roadmap for each model.

Approx. Cell Decomposition	**Star-Shaped Roadmaps**
Decomposition of $\mathcal{C}$ into *empty*, *full* and *mixed* cells	Decomposition of $\mathcal{C}$ into regions satisfying star-shaped property
Conservative approximation of $\mathcal{F}$	Complete connectivity of $\mathcal{F}$; every point in $\mathcal{F}$ is captured implicitly
Need to subdivide mixed cells	Not necessary to subdivide mixed regions that satisfy the star-shaped property
Large storage and search requirements; function of resolution parameter	Storage and search varies based on free space complexity; Lower requirements
Check for paths through empty cells and not mixed cells.	Check for paths through empty regions as well as mixed regions that satisfy the star-shaped property

TABLE II

Comparison: This table compares a number of aspects of our approach with approximate cell decomposition.

VII. Comparison and Discussion

In this section, we compare our approach with some prior approaches. We also discuss certain aspects of our motion planning algorithm.

A. Comparison with Prior Approaches

Our algorithm performs adaptive subdivision similar to cell decomposition algorithms [6]. However, there is one major difference; unlike exact cell decomposition methods, we do not compute an explicit decomposition of the free space. Instead, we compute a subdivision of the entire configuration space, which represents the free space implicitly. The main advantage of our approach is that we are able to perform the subdivision without an explicit representation of the free space. Most practical algorithms are based on approximate cell decomposition algorithms, which try to find a path through empty cells in the configuration space. By definition, the empty cells lie in the free space and result in a conservative approximation of the free space. We perform a detailed comparison with approximate cell decomposition algorithm in Table II. While approximate cell-decomposition algorithms are *resolution complete*, our algorithm is able to perform complete motion planning. One important benefit of our approach is that we do not always have to subdivide the mixed regions. If a mixed region satisfies the star-shaped test, then we do not subdivide it. We can plan paths through mixed cells directly by exploiting the star-shapedness property – this reduces the total number of subdivisions considerably making our approach practical for high-dof robots. Most applications of approximate cell decomposition have been limited to robots with three or four dof.

We generate samples in the free space that are represented by guards and connectors. Table III compares our deterministic sampling approach with randomized sampling approach by showing the different steps of the two approaches. Our approach does not need to perform explicit local planning to connect nearby samples. The star-shaped property ensures that the connectors link guards belonging to adjacent regions thus providing local planning implicitly. The main benefit of PRM-based methods is that they easily extend to very high dof robots, whereas our approach has additional overhead in terms of adaptive subdivision and conservative star-shaped tests.

Our approach shares some similarities with the visibility based probabilistic roadmap method (Visibility-PRM) [11]. Visibility-PRM method takes inter-sample visibility into account during the randomized sampling process. While the star-shaped property is related to visibility, it is different from the type of visibility computed by [11]. While the star-shaped property implicitly determines the visibility of an entire region, the visibility-PRM method computes the visibility of a new randomly computed sample with respect to the current set of samples. Finally, the goals of the two methods are very different: The objective of the visibility-PRM method is to generate a probabilistic roadmap with fewer nodes, whereas our goal is to do complete path planning.

Randomized Sampling	**Our Algorithm: Deterministic Sampling**
Compute samples randomly	Compute guards & connectors deterministically
Check whether samples are in free space	The guards and connectors are in free space by construction
Perform local planning between nearby samples	No *explicit local planning*; star-shaped property guarantees local collision-free paths
Easily extends to high-dof robots	Storage complexity and cost of star-shaped tests increases with number of dof
May not terminate with narrow passages or no collision-free path	Guaranteed to terminate if there are no tangential contacts in free space

TABLE III

Comparison: This table compares the steps of our approach with those of the randomized sampling approach.

Our current work builds on our prior work on isosurface extraction and translational motion planning [14], [13]. Our previous motion planning algorithm was limited to translational dof and used *complex cell* and star-shaped tests. Our new approach is relatively simpler and uses only the star-shaped test. Overall, the star-shaped roadmap based sampling is less conservative, easily extensible to higher dimensional configuration spaces with translations and rotational dof and less prone to degeneracy.

Our current work shares some similarities with the recent work of Delanoue et al. [4], which was developed independently. Their work is aimed at proving topological properties such as connectedness of sets. Their approach uses the star-shaped property to check if a set defined by a collection of non-linear inequalities is path-connected. Delanoue et al's current results are for two-dimensional sets defined by a few non-linear constraints. It is not clear whether their approach has been applied to path planning. The focus of our work is different – to use the star-shaped property to perform deterministic sampling for complete motion planning.

Model	Complexity			Performance			Statistics	
	Robot	Obstacle	# Surf	Subdivision & Guard (s)	Connector (s)	Planning (s)	# Guards	# Connectors
Gears 1	36	72	3,929	62	49	0.22	6,764	11362
Gears 2 (No path)	36	72	3,929	58	32	0.18	3,412	5,348
Assembly	224	224	256	10.1	5.8	0.22	6137	15,399
3R 1	3	32	140	12.3	4.9	0.43	11,349	30,566
3R 2 (No path)	3	32	176	12.2	4.4	0.14	10,062	25,270

TABLE I

Performance: *This table highlights the performance of our algorithm on different models. The model complexity is provided in terms of the size of the robot and the obstacle as well as the number of contact surfaces. The size of an object refers to the number of vertices for the planar* Gears *example and the number of triangles for the 3D* Assembly *example. The performance is measured in terms of the roadmap construction time and the time to answer a single planning query. The roadmap construction time is the sum of the time taken to compute an adaptive subdivision (includes guard computation) and the time to compute the connectors. The table also provides statistics on the number of guards and connectors in the roadmap.*

B. Discussion

The deterministic sampling algorithm presented in Sec. V performs an adaptive subdivision of the configuration space. At each step, we subdivide a region into sub-regions and different types of subdivision strategies could be employed. We could subdivide a region into d^2 equal sized regions where d is the dimension of the configuration space. Another alternative would be to perform a d-dimensional tetrahedral or simplicial subdivision of the region. We could also randomly select a point in the interior of the region and subdivide the region into tetrahedral regions with the point as an apex.

Our adaptive subdivision algorithm automatically performs additional subdivisions in the vicinity of the narrow passages in the free space. The number of subdivisions depends on the width of the narrow passages. Asymptotically speaking, the number of subdivisions is proportional to the log of the width. Our star-shaped roadmap algorithm captures the connectivity through the narrow passages and consequently, we are able to find a path without generating a very large number of samples or subdivisions. At the same time, our algorithm is able to terminate early if there is no collision free path.

C. Limitations

Our algorithm assumes that the free space does not have any tangential contacts. Hence it cannot handle cases where the robot must touch an obstacle in order to pass through a narrow passage to get to the goal configuration. A related limitation of our approach is that it does not support motion in the contact space – the robot is not allowed to touch any of the obstacles during its motion.

Our star-shaped test based on linear programming and interval arithmetic is conservative – the free space within a region may be star-shaped, but the region may not satisfy our test and hence may be subdivided unnecessarily.

VIII. Conclusions and Future Work

We have presented a simple approach for complete motion planning that relies on computing a star-shaped roadmap of the free space. We construct this roadmap using deterministic sampling. We show that the star-shaped roadmap generated by our algorithm captures the connectivity of the free space enabling us to perform complete path planning. Our approach is simple to implement and primarily relies on a star-shaped test which can easily be implemented. We have demonstrated the performance of our planner in complex scenarios with low dof robots. Our preliminary results are encouraging.

There are many avenues for future work. We are interested in the application of our algorithm to higher DOF motion planning. Our approach uses linear programming and interval arithmetic. Both these techniques are extensible to higher dimensional spaces. We would like to combine our approaches with randomized sampling techniques in order to generate better subdivisions for high-dof robots. We would also like to handle cases, where the robot is allowed to be in contact with the boundary of the obstacle.

References

[1] N. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In P. K. Agarwal, L. E. Kavraki, and M. Mason, editors, *Workshop on Algorithmic Foundations of Robotics*. A. K. Peters, Wellesley, MA, 1998.

[2] J. Basch, L. J. Guibas, D. Hsu, and A. Nguyen. Disconnection proofs for motion planning. In *International Conference on Robotics and Automation*, pages 1765–1772, 2001.

[3] J. Canny. *The Complexity of Robot Motion Planning*. ACM Doctoral Dissertation Award. MIT Press, 1988.

[4] N. Delanoue, L. Jaulin, and B. Cottenceau. Using interval arithmetic to prove that a set is path-connected. *Theoretical Computer Science, Special issue: Real Numbers and Computers.*, 2004.

[5] L. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, pages 12(4):566–580, 1996.

[6] J. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[7] J. Latombe. Motion planning: A journey of robots, molecules, digital actors, and other artifacts. *International Journal of Robotics Research*, pages 1119–1128, 1999.

[8] T. Lozano-Pérez and M. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Comm. ACM*, 22(10):560–570, 1979.

[9] J. O'Rourke. *Art Gallery Theorems and Algorithms*. The International Series of Monographs on Computer Science. Oxford University Press, New York, NY, 1987.

[10] J. T. Schwartz and M. Sharir. On the piano movers probelem ii, general techniques for computing topological properties of real algebraic manifolds. *Advances of Applied Maths*, 4:298–351, 1983.

[11] T. Simeon, J. P. Laumond, and C. Nissoux. Visibility based probabilistic roadmaps for motion planning. *Advanced Robotics Journal*, 14(6), 2000.

[12] J. M. Snyder. Interval analysis for computer graphics. In E. E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 121–130, July 1992.

[13] G. Varadhan, S. Krishnan, T. V. N. Sriram, and D. Manocha. A simple algorithm for complete motion planning of translating polyhedral robots. In *Workshop on Algorithmic Foundations of Robotics*, 2004.

[14] G. Varadhan, S. Krishnan, T. V. N. Sriram, and D. Manocha. Topology preserving surface extraction using adaptive subdivision. In *Eurographics Symposium on Geometry Processing*, 2004.

[15] G. Varadhan and D. Manocha. Star-shaped roadmaps - a deterministic sampling approach for complete motion planning. *UNC Technical Report TR05-001*, URL: http://gamma.cs.unc.edu/motion/, 2005.

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