

Path Planning for Ellipsoidal Robots and General Obstacles via Closed-Form Characterization of Minkowski Operations

Sipu Ruan¹, Qianli Ma², Karen L. Poblete¹, Yan Yan³, Gregory S. Chirikjian¹

¹ Laboratory for Computational Sensing and Robotics
Johns Hopkins University, Baltimore MD 21218, USA,
{ruansp, kpoblet1, gchirik1}@jhu.edu,

² Aptiv, Pittsburgh PA 15238, USA, qianli.b.ma@aptiv.com

³ Amperity Inc, Seattle WA 98134, USA, yanyan.jhu@gmail.com

Abstract. Path planning has long been one of the major research areas in robotics, with PRM and RRT being two of the most effective path planners. Though they are generally very efficient, these two sample-based planners can become computationally expensive in the important special case of narrow passage problems. This paper develops a path planning paradigm which uses ellipsoids and superquadrics to respectively encapsulate the rigid parts of the robot and obstacles. The main benefit in doing this is that configuration-space obstacles can be parameterized in closed form, thereby allowing prior knowledge to be used to avoid sampling infeasible configurations, in order to solve the narrow passage problem efficiently. Benchmark results for single-body robots show that, remarkably, the proposed method outperforms the sample-based planners in terms of the computational time in searching for a path through narrow corridors. Feasible extensions that integrate with sample-based planners to further solve the high dimensional multi-body problems are discussed, which will require substantial additional theoretical development in the future.

1 Introduction

Sample-based planners such as PRMs [1] and RRTs [2] (and a multitude of their extensions, e.g [3, 4]) have demonstrated remarkable success. These methods usually are based on polyhedral representations of robots and obstacles and perform explicit collision detection to assess the feasibility of samples. These methods have had a profound impact both within robotics and across other fields such as molecular docking, urban planning, and assembly automation.

It is well known that despite the great success of these methods, the “narrow passage problem” remains a significant challenge. The reason is that, generally speaking, sample-based approaches use a strategy of sampling states in the whole configuration space, followed by collision checking. When a robot and an obstacle are found to be in collision, the corresponding sample is discarded. Then, valid state configurations are connected by edges, where each edge is sub-sampled

and collision checking is done along the edge. If any of the states on the edge corresponds to a collision, the whole edge is discarded. This approach works extremely well when the obstacles in the environment are sparse. But when there is a narrow passage, an inordinate amount of computational time is spent on the samples and edges that eventually will be discarded. To increase the probability of sampling valid configurations in a narrow passage, various methods have been proposed such as [5–8]. However, there is still no guarantee of finding valid vertices efficiently within the corridor due to the probabilistic nature of sampling and collision checking. Therefore the first goal of this paper is:

1. Extend our previous method of parameterizing a priori the free space [9], and develop a planner so as to avoid traditional collision checking computations.

In the planner developed here, the basic objects are unions of ellipsoids for the robot and superquadrics for environmental features. One reason to use superquadrics is that they can characterize a wide range of the complex shapes while requiring only a few parameters [10–12].

It is well known that for a single rigid body with a fixed orientation in n -dimensional space, a “slice” of the configuration space (C-space) obstacle corresponding to this orientation is the Minkowski sum of the rigid body and the obstacles in the workspace [13–17], denoted as a “C-layer” [18, 19]. There is substantial literature on the computational complexity of Minkowski sums of polyhedra and faceted approximations of ellipsoids [20–24]. Recently an exact closed-form formula for n -dimensional ellipsoids was introduced and discussed [25]. As a generalization of that, the closed-form Minkowski sums of an ellipsoid and an arbitrary convex differentiable shape embedded in n -dimensional Euclidean space is presented here, with superquadrics being a typical example. This is another essential reason for the choice of superquadrics objects as environmental features in our new planner.

Minkowski sums characterize the C-space obstacles for the individual rigid components in an articulated robot, and the feasibility of a robot’s configuration corresponds to each rigid component of the robot in the complement of the union of C-space obstacles. Consequently collision-free samples can be generated. However, if one seeks to *connect* such samples using current sample-based planning paradigms like PRM or RRT, then collision checking is still required. Therefore, the second goal of this paper is:

2. Develop guaranteed safe and efficient methods for connecting configurations with different rotational components without performing traditional collision checking.

This applies the idea of the Kinematics of Containment for convex objects [26], as well as detailed derivations for ellipsoids [27, 28]. It is shown in this paper that one can enclose the robot by a slightly larger ellipsoid, so the allowable motions of the robot being fully contained can be characterized as a convex polyhedron, which is denoted by the “Local C-space” for a specific configuration.

As an evaluation, the performance of the proposed planner is compared with the sample-based algorithms from the well-known “Open Motion Planning Li-

brary (OMPL)” [29]. Path planning for planar motions, i.e. those on $SE(2)$, are demonstrated, with the robot being an ellipse and obstacles being superellipses.

The rest of the paper is organized as follows. Section 2 reviews the concepts of Minkowski sum and difference, and extends the closed-form expressions for Minkowski operations of an ellipsoid and a convex differentiable surface embedded in n -dimensional Euclidean space so that the free space of individual robot components can be parameterized. Section 3 reviews our previously proposed Highway RoadMap algorithm for elliptical robot and obstacles, and extends the idea to the ellipse-superellipse case. An idea of the “Local C-space” is then proposed to connect configurations with different rotational components. We then evaluate the efficiency and effectiveness of our algorithm on $SE(2)$ by benchmarking with probabilistic planners from OMPL, especially for the “narrow passage” problem, in Section 4. We conclude with discussions in Section 5.

2 Mathematical Preliminaries

This section provides the mathematical preliminaries for the path planning algorithm. At first, closed-form Minkowski sum and difference between an ellipsoid and any surface with implicit and parametric expressions is derived. As a concrete representation of the environment, a surface in the form of a superellipse is studied, where the closed-form Minkowski sum and difference between an ellipse and a superellipse is derived explicitly.

2.1 Minkowski Sum and Difference between Two Convex Objects

The Minkowski sum and difference of two convex point sets (or bodies) each centered at the origin, P_1 and P_2 in \mathbb{R}^n , are defined respectively as [30]

$$P_1 \oplus P_2 \doteq \{p_1 + p_2 \mid p_1 \in P_1, p_2 \in P_2\}, \text{ and } P_1 \ominus P_2 \doteq \{p \mid p + P_2 \subseteq P_1\}. \quad (1)$$

Alternatively, the Minkowski difference of two convex bodies can be defined relative to the Minkowski sum as the body $P'_1 = P_1 \ominus P_2$ where $P_1 = P'_1 \oplus P_2$. While it is relatively simple to mathematically define the Minkowski operations, calculating useful representations of Minkowski sums or differences can be difficult and computationally expensive, especially when the boundary of these regions requires an explicit representation.

2.2 Closed-form Minkowski Operations between an Ellipsoid and a General Convex Differentiable Surface Embedded in \mathbb{R}^n

It has been observed that the Minkowski sum and difference between two ellipsoids can be computed in closed-form. The computational procedure can be further extended when one ellipsoid is substituted by an arbitrary convex differentiable surface embedded in n -dimensional Euclidean space (\mathbb{R}^n).

Assume that S_1 is a hyper-surface embedded in \mathbb{R}^n , with implicit and parametric forms being

$$\Phi(\mathbf{x}_1) = 1 \quad \text{and} \quad \mathbf{x}_1 = \mathbf{f}(\boldsymbol{\psi}), \quad (2)$$

where $\Phi(\cdot)$ is a real-valued differentiable function of $\mathbf{x}_1 \in \mathbb{R}^n$ and \mathbf{f} is a differentiable n -dimensional vector-valued functions of $\boldsymbol{\psi} \in \mathbb{R}^{n-1}$. Let E_2 be an arbitrary ellipsoid in \mathbb{R}^n , with semi-axis lengths $\mathbf{a}_2 = [a_1, a_2, \dots, a_n]^\top$. Then, the implicit and explicit equations are of the form $\mathbf{x}^\top A_2^{-2} \mathbf{x} = 1$ and $\mathbf{x} = A_2 \mathbf{u}(\boldsymbol{\psi})$, where $A_2 = R_2 \Lambda(\mathbf{a}_2) R_2^\top$ is the shape matrix of E_2 where $R_2 \in \text{SO}(n)$ denotes the orientation of the ellipsoid, and $\Lambda(\cdot)$ is a diagonal matrix. Here $\mathbf{u}(\boldsymbol{\psi})$ is the standard parameterization of the n -dimensional hyper-sphere with parameters $\boldsymbol{\psi} = [\psi_1, \psi_2, \dots, \psi_{n-1}]^\top$.

The affine transformation that shrinks the ellipsoid into a sphere with radius $r = \min\{a_1, a_2, \dots, a_n\}$ on the surface S_1 can be expressed as $\mathbf{x}' \doteq T\mathbf{x}$, where $T = R_2 \Lambda(r/\mathbf{a}_2) R_2^\top$ denotes the “shrinking” affine transformation, which is symmetric and positive definite since $\Lambda(r/\mathbf{a}_2)$ is diagonal and positive definite.

The implicit expression for the “shrunk” S_1 , denoted as S'_1 , is $\Phi(T^{-1}\mathbf{x}') = 1$. Then the Minkowski sum between S'_1 and E'_2 (now is a sphere), is obtained by computing the boundary of the offset surface with offset radius r as $\mathbf{x}_{ofs} = \mathbf{x}' + r\mathbf{n}'$, where $\mathbf{n}' = \frac{\nabla\Phi(T^{-1}\mathbf{x}')}{\|\nabla\Phi(T^{-1}\mathbf{x}')\|}$ is the outward normal of the surface and $\nabla\Phi(T^{-1}\mathbf{x}') = T^{-\top} \nabla\Phi(\mathbf{x})$ with $T^{-\top} = (T^{-1})^\top = (T^\top)^{-1} = T^{-1}$.

The Minkowski sum between the original surface S_1 and ellipsoid E_2 can be given by “stretching” the transformed space back, using inverse affine transformation, as

$$\mathbf{x}_{eb} = T^{-1}\mathbf{x}_{ofs} = T^{-1}(T\mathbf{x} + r \frac{T^{-\top} \nabla\Phi(\mathbf{x})}{\|T^{-\top} \nabla\Phi(\mathbf{x})\|}) = \mathbf{x} + r \frac{T^{-2} \nabla\Phi(\mathbf{x})}{\|T^{-1} \nabla\Phi(\mathbf{x})\|} \quad (3)$$

The Minkowski difference $S_1 \ominus E_2$ therefore can be obtained by switching the plus sign in Eq. (3) to minus. However, for the Minkowski difference, a “curvature constraint” should be satisfied: after the “shrinking” operation, the curvature of every point on the transformed surface S'_1 should be smaller than the curvature of the transformed ellipsoid E'_2 .

2.3 Explicit Expressions of the Closed-form Minkowski Operations between an Ellipse and a Superellipse

We now give a concrete example for the closed-form Minkowski operations when the surface is a 2D superellipse. This formulates a mathematical representation for the implementation in this paper.

The implicit and explicit equations for a superellipse S_1 in \mathbb{R}^2 are defined as

$$\Phi(\mathbf{x}) = \left(\frac{x_1}{a_1}\right)^{\frac{2}{\epsilon}} + \left(\frac{y_1}{b_1}\right)^{\frac{2}{\epsilon}} = 1, \quad \text{and} \quad \mathbf{x} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} a_1 \cos^\epsilon \theta \\ b_1 \sin^\epsilon \theta \end{pmatrix}, \quad -\pi \leq \theta \leq \pi, \quad (4)$$

respectively. The shape described by the above function changes with ϵ . We only consider the case of $0 < \epsilon < 2$ to ensure the convexity of the corresponding

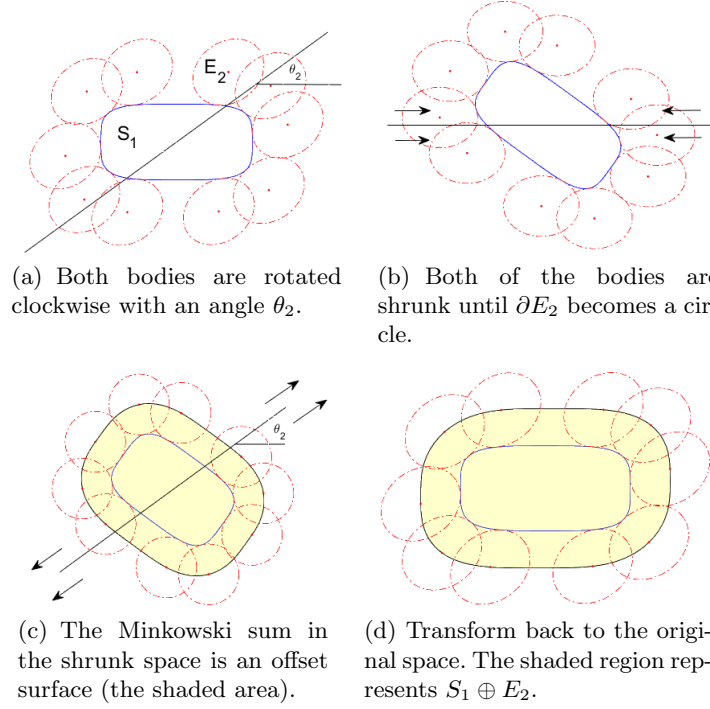


Fig. 1. Algorithm for obtaining the characterizations of the Minkowski sum between a superellipse S_1 and an ellipse E_2 .

shape. Then the normal vector of the superellipse can be obtained by calculating the gradient of $\Phi(x_1, y_1)$ as $\nabla\Phi(x_1(\theta), y_1(\theta)) = \frac{2}{\epsilon} \begin{pmatrix} \cos^{2-\epsilon} \theta / a_1 \\ \sin^{2-\epsilon} \theta / b_1 \end{pmatrix}$.

Suppose an ellipse is defined by the parameters $\mathbf{a}_2 = [a_2, b_2]^\top$ and its orientations be characterized as $R_2 = R(\theta) \in \text{SO}(2)$. If we further define $r = \min\{a_2, b_2\}$, then the “shrinking” transformation can be computed as $T = R_2 \Lambda(r/\mathbf{a}_2) R_2^\top$. Now we have all the information to calculate the closed-form Minkowski sum and difference between an ellipse and a superellipse, i.e. $S_1 \oplus E_2$ and $S_1 \ominus E_2$ respectively. Fig. 1 illustrates the computational process.

3 The Highway RoadMap Path Planning Algorithm for Ellipsoidal Robots: Reviews and Extensions

With the definition of the exact closed-form contact boundary of an ellipsoid and a superquadrics obtained in Sec.2.3, path planning problems can be solved effectively. In this section, the “Highway RoadMap” algorithm [9, 31] which takes advantage of the knowledge of collision-free C-space a priori is briefly reviewed.

Algorithm 1: Highway RoadMap Algorithm

Input: *Robot; Obstacle; Arena; EndPoint*
Output: Graph; Path

```

1 Rot = Discretized Orientations;
2 foreach Rot do
3    $C_{Obs}, C_{Arena} \leftarrow$  Minkowski Operations between Robot and Obstacle, Arena;
4    $Cell_{CF} \leftarrow$  Cell Decomposition by Sweep-Line Process;
5   Graph  $\leftarrow$  Vertex Generations and Edge Connections Within the C-Layer;
6 end
7 Graph  $\leftarrow$  Connect Closest vertices among Adjacent C-Layers;
8 Path  $\leftarrow$  Graph Search Process;
```

An extension for configuration connections with different rotational components is developed.

3.1 Overview of the Highway RoadMap Algorithm

The Highway RoadMap system is built based on the idea of cell decomposition at each orientation of the robot. At first, the orientations are discretized from the configuration space. Then, at each fixed orientation, a subset of the C-space that only contains translational motions is built, denoted as a “C-layer”. The roadmap system is built by first decomposing each C-layer into disjoint collision-free cells, and connecting vertices between adjacent C-layers. At each C-layer, The closed-form Minkowski sum and difference are computed between the robot and the obstacles and arenas, respectively ⁴. Note that for the sake of simplicity, one can always make the arena larger enough in comparison to the robot, so that the “curvature constraint” of Minkowski difference is not activated. Once the Minkowski operations are applied, the configuration space with C-obstacles is constructed, then the free space can be characterized and decomposed into disjoint collision-free cells. A subset of the roadmap can be constructed by detecting the middle point at each boundary of the adjacent cells as a vertex and connecting edges between two vertices. The entire roadmap system can then be constructed by connecting vertices among adjacent C-layers.

This procedure is based on our previously published work which uses two ellipsoids [9]. In this paper, we extend this description by using superquadrics, and propose a new method to connect vertices among different C-layers. The planning algorithm is illustrated in Algorithm 1.

3.2 A Sweep-Line Process for the Cell Decomposition within One C-Layer

Within one C-layer, the motion of a robot is restricted to translations only, and the Minkowski operations are applied here to obtain C-space obstacles and are-

⁴ Here the word “arena” denotes the bounded area in which the robot and obstacles are contained.

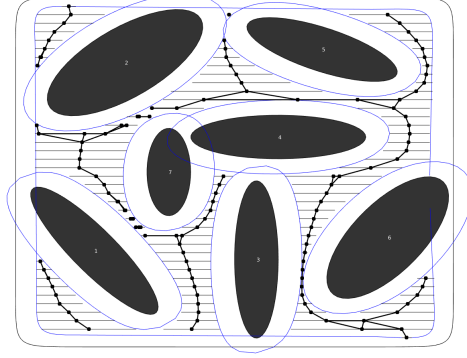


Fig. 2. The cell decomposition at one C-layer. Black superelliptical objects: obstacles; blue curves: the closed-form Minkowski sum and difference boundaries; thin black line segments: the sweep lines parallel to x-axis; black dots: collision free vertices within the C-layer; thick black line segments: collision free edges that connect vertices.

nas. To address this, a process similar to the “sweep line” method to decompose the collision-free space as disjoint cells has been proposed [32].

Suppose that the robot is constructed by a finite union of M ellipsoids E_1, E_2, \dots, E_M , and the transformations between the first ellipsoid E_1 and other ellipsoids E_2, E_3, \dots, E_M are defined as g_2, g_3, \dots, g_M respectively. Let the collision-free C-space for each ellipsoid E_i be denoted as C_i ($i = 1, \dots, M$). Then the collision-free space for the whole robot can be characterized as a union of them as viewed in the body frame of first ellipsoid, i.e. $C = C_1 \cap (g_2 \cdot C_2) \cap (g_3 \cdot C_3) \cap \dots \cap (g_M \cdot C_M)$, where $g_i \cdot C_j \doteq R_i C_j + \mathbf{t}_i$ ($i, j = 1, \dots, M$).

In order to detect those regions, a set of sweep lines parallel to the x-axis are generated. Each sweep line intersects with all the curves, with the intersecting points saved as pairs or intervals. Denoting the line segments within the obstacles as P_{O_i} , and those within the arenas as P_{A_i} , the collision-free line segment P_{CF} for each sweep line can be represented as

$$P_{CF} = \bigcap_{i=1}^{M_A \times M} P_{A_i} - \bigcup_{j=1}^{M_O \times M} P_{O_j} \quad (5)$$

where M_A and M_O are numbers of superquadrics that represent arenas and obstacles respectively. Figure 2 shows the decomposed C-space at one layer with collision-free cells highlighted by horizontal raster lines.

3.3 A Local Planner for Vertex Connections between C-Layers

Since each C-layer only represents one orientation of the robot, one must connect the subgraphs among different C-layers so that the robot can transform between different layers by rotations. It is always beneficial to find a continuous collision-free space that can enclose all the steps along the edge between two vertices.

Therefore, we propose a local planner to build a continuous convex C-space that allows the robot to move between two configurations in different C-layers.

The basic idea is to enclose the actual robot within a slightly larger ellipsoid, i.e. scale the actual robot up by a factor of $\epsilon = 0.1$. Then the robot is allowed to move small amounts inside the larger ellipsoid without collisions. Such motions can be described locally in the C-space, denoted as “Local C-space”. The local C-space becomes collision-free if the Minkowski operations are performed using the larger ellipsoid, and the descriptions of the local C-space can be done before building the roadmap a priori. Once the local C-space of the two vertices intersect, a new vertex can be generated within the intersecting area and connected to the two vertices. The following subsections introduce in detail the characterizations of the local C-space and the procedure to connect two vertices by a collision-free path.

Characterization of the Local C-space We first review the allowable motions of the smaller ellipsoid inside a larger ellipsoid, both of which can be described in the n -dimensional case. The related study traces back to the concept of the “Kinematics of Containment”, which deals with any convex body that moves inside another [26]. A recent study of a special case, in which both bodies are n -dimensional ellipsoids, can be applied here [27].

Given two n -dimensional ellipsoids E_a and E_b , with $E_a \subseteq E_b$, we let $\mathbf{a} = [a_1, a_2, \dots, a_n]^\top$, $\mathbf{b} = [b_1, b_2, \dots, b_n]^\top \in \mathbb{R}^n$ denote the semi-axes of E_a and E_b respectively. By substituting the explicit expression of the moving ellipsoid E_a into the implicit expression of the fixed ellipsoid E_b that is aligned with the world frame, the algebraic condition for E_a to move inside E_b without collision can be written as

$$(R_a \Lambda(\mathbf{a})\mathbf{u} + \mathbf{t}_a)^\top \Lambda^{-2}(\mathbf{b})(R_a \Lambda(\mathbf{a})\mathbf{u} + \mathbf{t}_a) \leq 1. \quad (6)$$

This highly nonlinear expression can be simplified by a small angle approximation. If E_a is restricted to infinitesimal motions, the rotation part calculated by exponential map can be approximated to the first order as

$$R_a = \exp(\hat{\omega}_a) \approx \mathbb{I} + \hat{\omega}_a, \quad (7)$$

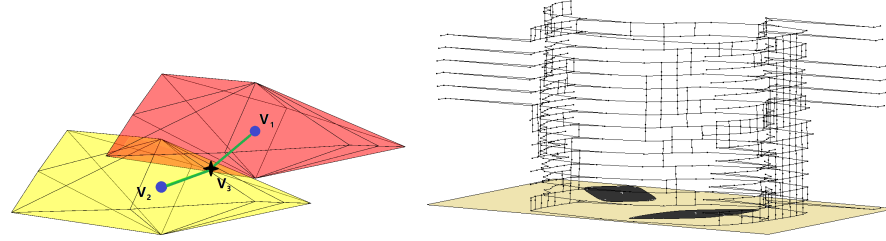
where $\hat{\omega}_a \in \mathfrak{so}(n)$. Grouping parameters (\mathbf{u}) and variables ($\boldsymbol{\omega}$ and \mathbf{t}) gives the first-order approximation of the left-hand side of the algebraic condition of containment as

$$C_{\mathbf{u}}(\boldsymbol{\xi}) = \boldsymbol{\xi}^\top H(\mathbf{u})\boldsymbol{\xi} + \mathbf{h}^\top(\mathbf{u})\boldsymbol{\xi} + c(\mathbf{u}), \quad (8)$$

where $H(\mathbf{u}) \in \mathbb{R}^{n(n+1)/2 \times n(n+1)/2}$, $\mathbf{h}(\mathbf{u}) \in \mathbb{R}^{n(n+1)/2}$ and $c(\mathbf{u}) \in \mathbb{R}$. The first order algebraic condition of containment is then defined as $C_{\mathbf{u}}(\boldsymbol{\xi}) \leq 1$.

It can be further proved that Eq. (8) is a family of convex functions, with parameters \mathbf{u} and unknown variables $\boldsymbol{\xi} \doteq [\boldsymbol{\omega}^\top, \mathbf{t}^\top]^\top \in \mathbb{R}^{n(n+1)/2}$.⁵ As a result,

⁵ Explicitly, $\boldsymbol{\omega} = \log^\vee(R)$, $R \in \text{SO}(n)$ and $\mathbf{t} \in \mathbb{R}^n$ (\mathbf{t} is the actual translation as seen in the world reference frame). And the pair (R, \mathbf{t}) forms the “Pose Change Group”, i.e. $\text{PCG}(n) \doteq \text{SO}(n) \times \mathbb{R}^n$, with the group operation being a direct product, which is different than $\text{SE}(n) \doteq \text{SO}(n) \ltimes \mathbb{R}^n$ (for more details, see [33]).



(a) Edges between C-layers in the “Local C-space”. (b) A complete Highway RoadMap in C-space.

Fig. 3. (a) A collision-free vertex connection scheme between adjacent C-layers. Blue dots are the two vertices, V_1 and V_2 , with convex polyhedron being their local C-space. The green line segments connect V_3 at the intersection and the two vertices respectively. (b) The fully connected graph structure. The vertical axis represents the rotational angle; black dots are valid vertices and line segments are collision-free edges.

enclosing several valid configurations by a convex hull gives a convex polyhedron in the C-space of the smaller ellipsoid. This convex polyhedron describes a collision-free subspace and any path inside is guaranteed to be collision-free. The convex polyhedron is a lower bound of the actual local C-space, where the first-order approximation works well when the rotational motion is small. Therefore, not only the inflation factor ϵ , but the roundness of the ellipsoid also affects the approximation. With the increase of the aspect ratio, which quantifies the roundness, the convex lower bound takes larger portion of the volume related to the actual local C-space. For more details about the performance of convex lower bound of the local C-space, see [28].

Vertex Connections Based on the Convex Polyhedron Local C-space

To further connect two vertices, V_1 and V_2 , by a collision-free path $Path_{12}$, one can first define a new middle vertex V_3 that is inside the intersection of the Local C-space of V_1 and V_2 . Then connecting V_1, V_3 and V_2, V_3 by line segments gives $Path_{13}$ and $Path_{23}$ respectively. These two path segments are guaranteed to be collision-free since both are fully inside the convex polyhedron of V_1 or V_2 . Finally, the desired collision-free path is a combination of the two segments, i.e. $Path_{12} = Path_{13} \cup Path_{23}$. Fig.3 demonstrates the proposed connection scheme for vertices in different C-layers, and a complete Highway RoadMap system in the C-space with valid vertices being connected.

4 Experiments on Path Planning for a 2D Elliptical Robot

As a demonstration of the extended Highway RoadMap algorithm, we consider a 2D single-body planning problem, where the robot is an ellipse and the arena and

Table 1. Benchmark Parameters for Highway RoadMap and Sample-Based Planners From OMPL.

Parameters	Explanations
Maps	Sparse, Cluttered, Maze
Samplers	Uniform, Obstacle-based(OB), Bridge
Shapes	Ellipse, SuperEllipse
N_L	Number of C-layers for Highway RoadMap
N_{SL}	Number of sweep lines for Highway RoadMap
N_V	Number of valid configurations
N_E	Number of valid edges that connect two valid configurations
N_P	Number of vertices in the collision-free path

obstacles are modeled as superellipses. To evaluate the performance of the proposed algorithm, benchmarks are performed with three commonly-used sample-based planners (PRM, RRT, RRT-Connect) from the well-known Open Motion Planning Library (OMPL). Moreover, three different sampling methods are also compared for the probabilistic planners: uniform random sampling (Uniform), obstacle-based sampling (OB) [6] and bridge test (Bridge) [7]. This section describes the benchmark process and results, followed by some discussions and potential extension in the future work. Our algorithm is implemented in C++ and all the comparisons are performed in an Intel Core i7 CPU at 3.60GHz.

4.1 Experimental Parameters

Our experimental parameters include two types of shapes for representing the obstacles and define different kinds of maps for benchmark environments. The obstacles are defined as ellipses and superellipses, and for each shape, three maps are considered. We deal with a sparse map, where there are only a few obstacles and the free space occupies a majority of the area; a cluttered map, where more space is occupied by obstacles which are placed in different orientations; and a maze map, where only some narrow passages are free. Those maps capture different real life scenarios the robot might face, which can evaluate the performance of difference algorithms. And the comparisons include the running time for solving the problem and the success rate across multiple experimental trials.

For each map and each algorithm, we perform 50 planning trials and compute the average planning time and success rate. Note that for the probabilistic algorithms, a time limit of 60 seconds is set for one planning trial, which means the planning fails if the time exceeds the limit. The list of parameters for our comparisons is shown in Table 1.

Table 2 provides the implementation details for the Highway RoadMap planner, where the numbers of C-layers and sweep lines and the resulting vertices, edges and valid vertices on the path are provided.

Table 2. Implementation details of the parameters for Highway RoadMap planner

Shape	Map	N_L	N_{SL}	N_V	N_E	N_P
Ellipse	Sparse	15	17	723	916	25
	Cluttered	20	35	3621	4828	37
	Maze	50	40	10636	14699	486
SuperEllipse	Sparse	14	10	493	672	21
	Cluttered	14	25	2009	2547	72
	Maze	55	30	9782	13450	554

4.2 Collision Checking Methods for the Sample-Based Algorithms

In sample-based methods, the majority of time is consumed in collision checking, so the choice of a relatively fast collision checker is a priority. The standard and widely-used collision checking library, “Flexible Collision Library (FCL)”, has been applied in many scenarios [34], therefore we include FCL in our benchmark process. In particular, a special and efficient ellipsoidal method from FCL is applied for the case of elliptical obstacles. Since the input parameters for describing the ellipses for the planners used in our experiments are the same, the comparison is fair. For the case of superellipses, the objects are treated as polygons, and traditional collision checking for polygons is applied. To make the checking process more efficient as for a relatively fair comparison, we characterize each superellipse object by 10 discrete points on its boundary and construct triangulation meshes as inputs for the polygonal collision checking. We select 10 points on the boundary because a superelliptical object can be arbitrarily characterized and the speed of collision checking is still fast.

4.3 Experimental Results

This section describes the experimental results for benchmarking, followed by some discussions and potential extensions for future work.

Figure 4 shows the maps used for benchmark as well as the valid path Highway RoadMap finds, which verifies the correctness of our implementation. As a comparison, Fig. 5 shows the valid paths found by the three sampled-based planners from OMPL. Different sampling methods are also compared, and shown at the bottom row. It is evident that either bridge or obstacle-based method samples more valid configurations inside the narrow corridor.

Figure 6 shows the planning time for different algorithms on different maps. The “errorbox” plots demonstrate the statistics of the planning time among the 50 trials of the experiments. Although in the sparse map the Highway RoadMap is almost twice as slow as the probabilistic planners, as the map becomes more dense and cluttered, its speed increases slightly and it starts to take the lead among other planners. It becomes more obvious in the maze map that the speed of Highway RoadMap is much faster than the RRT-connect planner, which is the most efficient among the three sample-based planners from the comparisons.

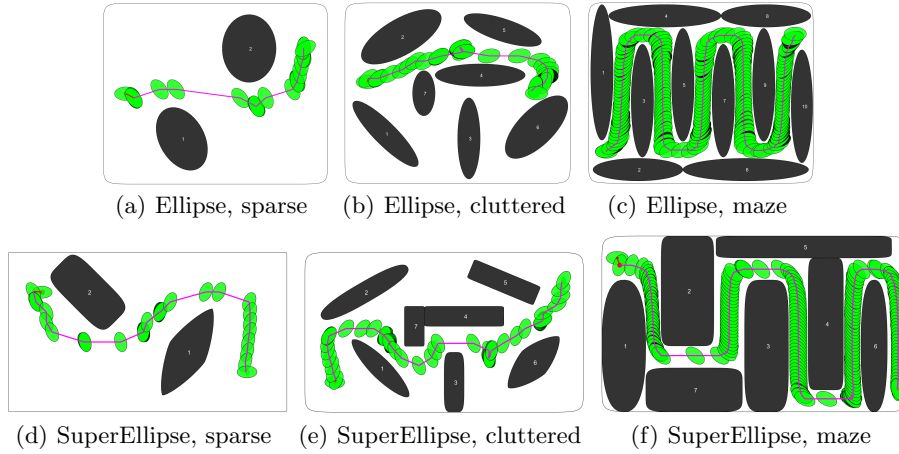


Fig. 4. The maps with elliptical and superelliptical obstacles for benchmarking the algorithms. The red and green ellipses are the robot at the start and goal configurations respectively.

Furthermore, because of deterministic nature of the Highway RoadMap, it behaves much more stable among different trials, while sample-based planners have relatively large variance. The choice of collision checking methods matters, since the sample-based planners becomes slower when the obstacles are superellipses and polygonal collision checking is applied. For the superellipse maze map, almost all the trials for the sample-based planners fail to find a solution within the time limit. Therefore we only show the planning time comparisons for the RRT-Connect planner, which illustrates that Highway RoadMap is more advantageous when solving the “narrow passage” problem by having a higher quantity of valid vertices in less computational time.

4.4 Discussion

The extended Highway RoadMap planner solves the path planning problems on $SE(2)$ in different scenarios efficiently. This is due to usage of a closed-form expression of Minkowski sum and difference that explicitly characterizes the C-space, in addition to the effectiveness of the cell decomposition method to solve the “narrow passage” problem. On the other hand, the sample-based planners are very efficient when the environment is sparse, but start to drop speed as the space occupied by obstacles increases. This is because most of the sampled configurations are discarded, so the “sample-check-discard-resample” process iterates much longer than in sparse environments.

The idea of “local C-space” gives a connection strategy for vertices on adjacent C-layers, which requires the existence of the intersection volume between the local C-space of the two vertices. In the $SE(2)$ case, a necessary condition for this existence requirement is that the “gaps” between adjacent C-layers should

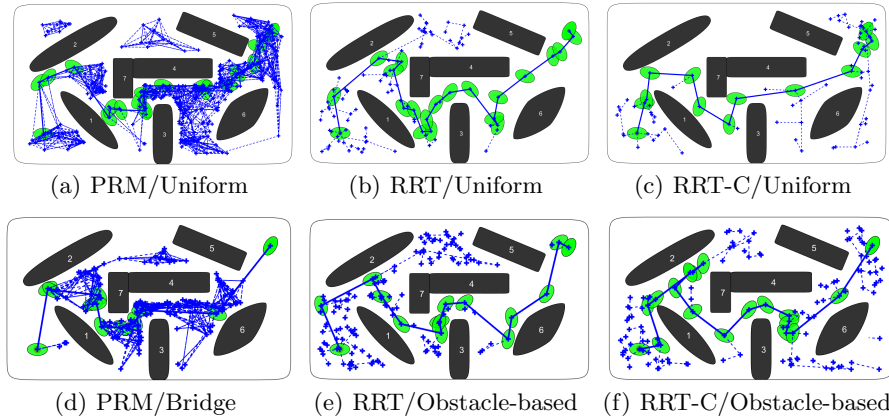


Fig. 5. A demonstration of the paths found by sample-based planners with different sampling strategies. The environment is the superellipse cluttered map. The blue dots and line segments are valid configurations and edges.

be smaller than the largest rotational angle, therefore there is a trade-off between selecting the inflation factor and the number of layers. The largest angle of rotation can be computed in closed-form, when fixing the smaller ellipse to the center of the larger one.

The major current limitation of the Highway RoadMap planner, however, is that its extension to solve high degree-of-freedom multi-body problems is not yet clear. The computational complexity of a naive implementation would grow exponentially as the dimension of the C-space increases. Addressing this problem is a challenge for future work.

One of the direct extensions to the current implementation is to solve the SE(3) planning problem, using a similar computational process. In fact, hybrid planners might be a promising solution, where one can either construct a whole graph structure by sampling in SO(3) and then generate valid vertices and edges in a similar fashion to the PRM method. A second option is to build a tree structure by repeatedly sampling orientations with associated C-layers until reaching the goal, similar to RRT. These possibilities potentially allow the Highway RoadMap planner to solve higher dimensional planning problems with narrow passages.

5 Conclusion

This paper proposes a closed-form characterization of Minkowski sums and differences between an ellipsoid and a general convex differentiable surface embedded in \mathbb{R}^n . As a specific demonstration, the Minkowski operations between a 2D ellipse and a superellipse are explicitly derived. These formulate the mathematical basis for describing the configuration space of an elliptical robot with superelliptical obstacles. With prior explicit knowledge of the C-space, a path planning

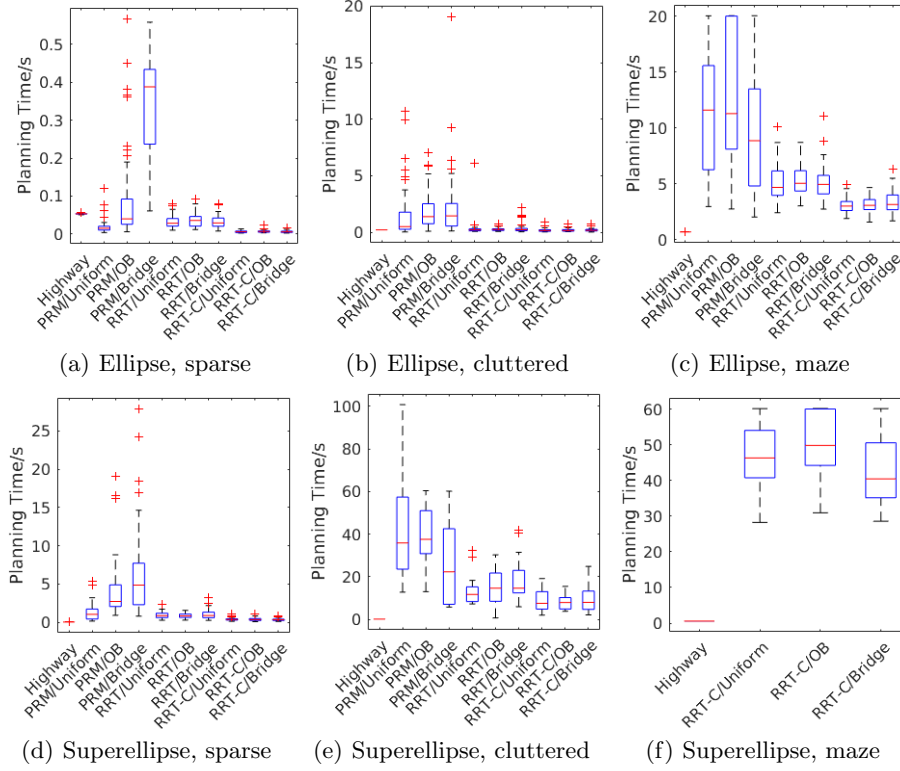


Fig. 6. Planning time comparisons between different algorithms.

problem can be solved by the proposed extended “Highway RoadMap” planner, where configurations with different rotational components can be connected via the procedure of generating the “Local C-space” for each configuration. A benchmark scheme with an elliptical robot and superelliptical obstacles is performed using the Highway RoadMap and PRM, RRT and RRT-Connect planners from the Open Motion Planning Library. Different sampling strategies for the narrow passage problem are also applied to sample-based planners. The results show that the extended Highway RoadMap outperforms the sample-based planners on the cluttered and maze maps, where obstacles occupy much space. Finally, potential extensions of the Highway RoadMap to higher dimensional planning problems with narrow passages are discussed, which the authors plan to explore more in the future. By combining the efficient explicit descriptions of the configuration space presented here and the effectiveness of the sample-based planners on high dimensional C-space, hybrid path planners can thereby potentially achieve better performance in higher dimensional cluttered environment.

Acknowledgements

The authors would like to thank Dr. Fan Yang, Mr. Thomas W. Mitchel and Mr. Zeyi Wang for useful discussions. This work was performed under National Science Foundation grant IIS-1619050 and Office of Naval Research Award N00014-17-1-2142. The ideas expressed in this paper are solely those of the authors.

References

1. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation* **12**(4), 566–580 (1996)
2. LaValle, S.M.: Rapidly-exploring random trees: A new tool for path planning (1998)
3. Kuffner, J.J., LaValle, S.M.: RRT-connect: An efficient approach to single-query path planning. In: *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2, pp. 995–1001. IEEE (2000)
4. Bohlin, R., Kavraki, L.E.: Path planning using lazy PRM. In: *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, pp. 521–528. IEEE (2000)
5. Varadhan, G., Manocha, D.: Star-shaped Roadmaps-A deterministic sampling approach for complete motion planning. In: *Robotics: Science and Systems*, vol. 173. Citeseer (2005)
6. Rodriguez, S., Tang, X., Lien, J.M., Amato, N.M.: An obstacle-based rapidly-exploring random tree. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 895–900. IEEE (2006)
7. Hsu, D., Jiang, T., Reif, J., Sun, Z.: The bridge test for sampling narrow passages with probabilistic roadmap planners. In: *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 3, pp. 4420–4426. IEEE (2003)
8. Shi, K., Denny, J., Amato, N.M.: Spark PRM: Using RRTs within PRMs to efficiently explore narrow passages. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 4659–4666. IEEE (2014)
9. Yan, Y., Ma, Q., Chirikjian, G.S.: Path planning based on closed-form characterization of collision-free configuration-spaces for ellipsoidal bodies obstacles and environments. In: *Proc. 1st Int. Workshop Robot Learn. Planning*, pp. 13–19 (2016)
10. Barr, A.H.: Superquadrics and angle-preserving transformations. *IEEE Computer graphics and Applications* **1**(1), 11–23 (1981)
11. Jaklic, A., Leonardis, A., Solina, F.: *Segmentation and recovery of superquadrics*, vol. 20. Springer Science & Business Media (2013)
12. Agba, E.I., Wong, T.L., Huang, M.Z., Clark, A.M.: Objects interaction using superquadrics for telemanipulation system simulation. *Journal of robotic systems* **10**(1), 1–22 (1993)
13. Guibas, L., Ramshaw, L., Stolfi, J.: A kinetic framework for computational geometry. In: *Foundations of Computer Science, 1983., 24th Annual Symposium on*, pp. 100–111. IEEE (1983)
14. Latombe, J.C.: *Robot motion planning*, vol. 124. Springer Science & Business Media (2012)

15. Kavraki, L.E.: Computation of configuration-space obstacles using the fast Fourier transform. *IEEE Transactions on Robotics and Automation* **11**(3), 408–413 (1995)
16. Agarwal, P.K., Flato, E., Halperin, D.: Polygon decomposition for efficient construction of Minkowski sums. *Computational Geometry* **21**(1), 39–61 (2002)
17. Nelaturi, S., Shapiro, V.: Configuration products and quotients in geometric modeling. *Computer-Aided Design* **43**(7), 781–794 (2011)
18. Lien, J.M.: Hybrid motion planning using minkowski sums. *Proceedings of robotics: science and systems IV* (2008)
19. Nelaturi, S., Shapiro, V.: Solving inverse configuration space problems by adaptive sampling. *Computer-Aided Design* **45**(2), 373–382 (2013)
20. Fogel, E., Halperin, D.: Exact and efficient construction of Minkowski sums of convex polyhedra with applications. *Computer-Aided Design* **39**(11), 929–940 (2007)
21. Hachenberger, P.: Exact Minkowski sums of polyhedra and exact and efficient decomposition of polyhedra into convex pieces. *Algorithmica* **55**(2), 329–345 (2009)
22. Behar, E., Lien, J.M.: Dynamic Minkowski sum of convex shapes. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3463–3468. IEEE (2011)
23. Kurzhanskiy, A.A., Varaiya, P.: Ellipsoidal Toolbox (ET). In: *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 1498–1503 (2006). DOI 10.1109/CDC.2006.377036
24. Hartquist, E.E., Menon, J., Suresh, K., Voelcker, H.B., Zagajac, J.: A computing strategy for applications involving offsets, sweeps, and Minkowski operations. *Computer-Aided Design* **31**(3), 175–183 (1999)
25. Yan, Y., Chirikjian, G.S.: Closed-form characterization of the Minkowski sum and difference of two ellipsoids. *Geometriae Dedicata* **177**(1), 103–128 (2015)
26. Chirikjian, G.S., Yan, Y.: The kinematics of containment. In: *Advances in Robot Kinematics*, pp. 355–364. Springer (2014)
27. Ma, Q., Chirikjian, G.S.: A closed-form lower bound on the allowable motion for an ellipsoidal body and environment. In: *Proc. 2015 IDETC/CIE*, pp. V05CT08A055–V05CT08A055. American Society of Mechanical Engineers (2015)
28. Ruan, S., Chirikjian, G.S., Ding, J.: Lower bounds of the allowable motions of one n-dimensional ellipsoid contained in another. In: *Proc. 2018 IDETC/CIE*, pp. V05BT07A081–V05BT07A081. American Society of Mechanical Engineers (2018)
29. Sucan, I.A., Moll, M., Kavraki, L.E.: The open motion planning library. *IEEE Robotics & Automation Magazine* **19**(4), 72–82 (2012)
30. Berg, M.d., Cheong, O., Kreveld, M.v., Overmars, M.: *Computational geometry: algorithms and applications*. Springer-Verlag TELOS (2008)
31. Yan, Y.: *Geometric motion planning methods for robotics and biological crystallography*. Ph.D. thesis (2014)
32. Choset, H.M., Hutchinson, S., Lynch, K.M., Kantor, G., Burgard, W., Kavraki, L.E., Thrun, S.: *Principles of robot motion: theory, algorithms, and implementation*. MIT press (2005)
33. Chirikjian, G.S., Mahony, R., Ruan, S., Trumpf, J.: Pose changes from a different point of view. *Journal of Mechanisms and Robotics* **10**(2), 021,008 (2018)
34. Pan, J., Chitta, S., Manocha, D.: FCL: A general purpose library for collision and proximity queries. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 3859–3866. IEEE (2012)