Proceedings of the 1st International Workshop on Robot Learning and Planning (RLP 2016)
in conjunction with 2016 Robotics: Science and Systems
June 18, 2016 – Ann Arbor, Michigan, USA

Path Planning Based on Closed-Form Characterization of Collision-Free Configuration-Spaces for Ellipsoidal Bodies, Obstacles, and Environments

Yan Yan[1], Qianli Ma[2], and Gregory S. Chirikjian[2]

***Abstract*—A closed-form parameterization of the collision-free configuration spaces (C-spaces) of robots and obstacles represented as finite unions of ellipsoids is presented. These objects can be quite general, including nonconvex bodies, and this approach represents an alternative to polyhedral representations of bodies. With this method, there is never any reason to sample and discard configurations suspected of being in collision, and existing sample-based planners can be modified to operate in areas of C-space that are *a priori* guaranteed to be collision-free. This all builds on the recent work on computing exact boundaries of Minkowski sums and differences of two arbitrary ellipsoids using affine transformations and the analytic properties of offset surfaces.**

A "highway" roadmap system is then constructed to connect the collision-free regions. Unlike other skeletal/roadmap decompositions of C-space, collision checking in this C-space graph can be eliminated not only for the vertices, but also for the edges, and the problem is simplified to a search for a connected path in an adjacency graph of the roadmap. We apply this approach of "knowing where to look" for path planning in C-spaces with narrow passages and demonstrate its potential by comparing with the well known sample-based path planning method that does not currently have the ability to take advantage of *a priori* knowledge of collision-free regions of C-space.

I. Introduction

Path planning is probably the most studied problem in robotics, and Probabilistic Road Map method (PRM) [1] and Rapidly-exploring Random Trees (RRT) [2] are the two of most successful algorithms so far. PRM is a multi-query planner which can solve different planning requests in the same map, whereas RRT is a single query planner which incrementally grows from the starting configuration towards to the goal configuration in a tree form. There are also many variants of both PRM and RRT, and the majority of the planners fall into the category of sampling based planning (SBP). A most recent review on SBP methods can be found in [3]. Basically, SBP methods randomly sample the C-space of the robot and to connect the collision-free sample nodes to form a feasible path. In contrast to most SBP methods which are probabilistic, there are also a number of planners that are deterministic such as [4, 5, 6]. SBP is easily extensible to robots with high degrees of freedom, whereas deterministic planners tend to be better at handling narrow passage problems. To take advantages of both types of motion planners, several hybrid planners are proposed [7, 8]. Specifically, [8] proposed a method named M-sum, which used point-based Minkowski operations [9] to calculate a number of "subspaces of the configuration space" (C-slices) for polyhedra in a random approach, and connect configurations within and among the obtained C-slices to form a roadmap. [10] also employed support vector machine (SVM) classification technique to efficiently approximate a high-dimensional configuration space to accelerate motion planning.

Minkowski operations, or equivalently Minkowski sum and difference, have broad applications in CAD/CAM, assembly planning [11] and computer-aided design [12]. In the context of robot motion planning, given an obstacle P_1 and a robot P_2 that can only translate, if one further defines a reference point on P_2, then $P_1 \oplus P_2$ will be the locus of the reference point where $P_1 \cap P_2 \neq \emptyset$. Similarly, if P_1 is an "arena" in which P_2 is translating, then $P_1 \ominus P_2$ is the locus of the reference point where $P_1 \cap P_2 = P_2$. The sum is called a C-space obstacle and the difference is called the P_2's collision-free C-space with respect to (w.r.t.) the arena. Some also view the above two spaces together as the "contact space". Despite the simple mathematical definition of Minkowski operations, it is well known that computing the representations of Minkowski sum or difference is very computationally intensive. Many algorithms have been proposed for numerically calculating the Minkowski sum between polygons or polyhedron in either 2D or 3D environment [12, 13, 14, 15, 16]. The algorithms either compute the convolution of geometric boundaries [14], or employ polygon/polyhedra decompositions [12, 15, 16, 17]. Compared to the above numerical approach, it is shown that the exact Minkowski sum and difference of two ellipsoids can be computed in closed form [18]. For the closed-form Minkowski sum operation, it can be applied to any ellipsoids at any arbitrary orientation in any dimensional Euclidean space, whereas the Minkowski difference operation can be used wherever it exists. The approaches are entirely analytical and in closed-form which is computationally efficient in nature.

Though ellipses/ellipsoids are relatively simple compared to polygons/polyhedra, they are widely applied in collision checking [19, 20, 21, 22]. Ellipsoids usually sever as the boundary for humanoids in motion planning [23] and are also used as the bounding volume for serial manipulators as in [22]. It also shows up in the real-time collision-free navigation between elliptical agents [24].

Due to the nice properties of ellipses/ellipsoids in computing the Minkowski sum/difference and in approximating geometric shapes, we develop a method based on ellipses/ellipsoids that can be used in conjunction with existing path planning algorithms, particularly sampling-based methods and roadmaps based on cell decompositions, to provide better performance in narrow-passage problems. The basic approach is to describe

[1] Dr. Yan Yan is with Amazon.com Inc. yanya.jhu@gmail.com
[2]Qianli Ma and Gregory S. Chirikjian are with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA mqianli1@jhu.edu, gregc@jhu.edu

two spaces parametrically in closed form: (1) The space of motions that is allowable for one ellipsoid (e.g., a single-rigid-body robot) to move without sharing any point with another ellipsoid (e.g., an obstacle); (2) the space of motions in which one ellipsoid (e.g., the robot) can move while being fully contained in another ellipsoid (e.g., the arena). Equipped with these tools, it becomes possible *a priori* to "throw away" vast volumes within C-space that correspond to robot/obstacle or robot/arena-boundary collisions by not generating samples in regions of C-space that are known not to be feasible. In this way, the computational time and storage requirements for sample-based methods are improved relative to sample-and-discard strategies that are not informed *a priori* about the features of the free space. Moreover, instead of generating large numbers of random samples in the collision-free C-space (the vast majority of which are discarded), the collision-free C-space can be directly decomposed into cells, and an adjacency graph can be constructed to encode the adjacency relationships of the cells, and the adjacency graph can therefore serve as a roadmap of the free space.

Several topics that have been studies in the literature previously may appear at a superficial level to be related to the current formulation. For example, the rapid numerical characterization of C-space obstacles as zero-one functions ('zero' corresponding to the free space and 'one' corresponding to obstacle regions) has been investigated [25, 26], as has the idea of marginalizing over C-space degrees of freedom and superimposing obstacles treated as permeable boundaries, with permeabilities added [27]. Though these numerical approaches characterize the free space, and provide the ability to rapidly answer the query as to whether a particular point in C-space is feasible or not, they are not instructive in characterizing the set of all feasible points in reasonable time, and they do not scale well with increased dimensionality of the C-space. In contrast, this paper presents a closed-form method for characterizing the free space of a relatively broad family of objects (ellipsoids). The finite union of these ellipsoids can be used to approximate any robot, obstacle, or environmental boundary either by inscription or circumscription, and the feasible part of the C-space can be described by appropriate unions and intersections of those for individual pairs of interacting ellipsoids.

The remainder of this paper is structured as follows. Sec. II reviews on the Minkowski sum and difference of two ellipsoids in closed forms. Sec. III integrates two aspects of the problem — parameterizing collision-free spaces of the robot and obstacle, and robot and environmental boundary. The intersection of the collision-free regions is detected with a line-sweeping algorithm and saved as collision-free intervals. Sec. IV proposes an approach to build a highway roadmap system based on the midpoints of the collision-free intervals along each sweep line. Also, the highway roadmap planner is demonstrated with path planning examples, in which all of the robot, the obstacles and the arena are constructed as the finite unions of ellipses/ellipsoids. By comparing the computational speed with some SBP methods, the efficacy and the potential, especially for the narrow passage problem, is demonstrated. Finally, Sec. V presents our conclusions.

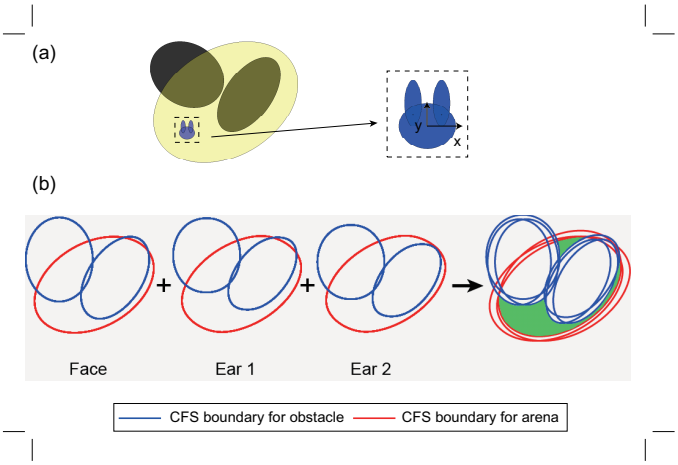

Fig. 1. (a) The example of one rabbit-shaped robot translating inside an elliptical arena that contains elliptical obstacles. (b) The boundaries of the collision-free space (CFS) for the robot-obstacle interaction (blue curves) and those for the robot-environment interaction (red curves) for the rabbit face and two ears, respectively. The CFS is illustrated as the green-shaded region.

II. Related Work on Closed-Form Characterization of Minkowski Operations of Two Ellipsoids

In this section, we briefly review on the closed-form Minkowski sum/difference on ellipses/ellipsoids in [18]. For convenience, we use the word "ellipsoid" to describe a 2D ellipse and also an n-dimensional hyper-ellipsoid when $n > 3$. Given two convex sets P_1 and P_2 in $\mathbb{R}^n$ each centered at the origin, the Minkowski sum is defined as

$$P_1 \oplus P_2 \doteq \{p_1 + p_2 \mid p_1 \in P_1, p_2 \in P_2\}, \tag{1}$$

and the Minkowski difference is defined as [28]

$$P_1 \ominus P_2 \doteq \bigcap_{p_2 \in P_2} (P_1 + p_2). \tag{2}$$

Define two arbitrary n-dimensional ellipsoids as E_1 and E_2 with the semi-axis lengths given by $\mathbf{a}_1 = [a_1, ..., a_n] \in \mathbb{R}^n$ and $\mathbf{a}_2 = [a'_1, ..., a'_n] \in \mathbb{R}^n$. Fix the center of mass at the origin and align the principal axes along the axes of the reference frame, ellipsoid E_1 will have the implicit and explicit equations as

$$\Phi(\mathbf{x}) \doteq \mathbf{x}^T \Lambda^{-2}(\mathbf{a}_1)\mathbf{x} = 1 \quad \text{and} \quad \mathbf{x} = \Lambda(\mathbf{a}_1)\mathbf{u}(\phi). \tag{3}$$

where $\Lambda(\mathbf{a}_1)$ is the $n \times n$ diagonal matrix with entry a_i at the location Λ_{ii}, $\Lambda^m(\mathbf{a}_1) = [\Lambda(\mathbf{a}_1)]^m$ is the m^{th} power of this matrix, and $\mathbf{u}(\phi)$ denotes the hyper-sphere S^{n-1} with $n-1$ angles $\phi = [\phi_1, ..., \phi_{n-1}]$. A rotated ellipsoid of the same shape and center of mass with ellipsoid E_1 has the implicit and explicit expressions as $\hat{\mathbf{x}}^T R_1 \Lambda^{-2}(\mathbf{a}_1) R_1^T \hat{\mathbf{x}} = 1$ and $\hat{\mathbf{x}} = R_1 \Lambda(\mathbf{a}_1)\mathbf{u}(\phi)$, where $R \in SO(n)$ is a $n \times n$ rotation matrix.

As with any convex set, the Minkowski sum of E_1 and E_2 is denoted as $E_1 \oplus E_2$. For the closed-form Minkowski sum of ellipsoids, the basic idea is to first apply an affine transformation onto E_1 and E_2 such that E_2 shrinks into a sphere of the radius $r = \min\{a'_1, a'_2, ..., a'_n\}$. After calculating the offset surface of the shrunk E_1 with radius r, the offset

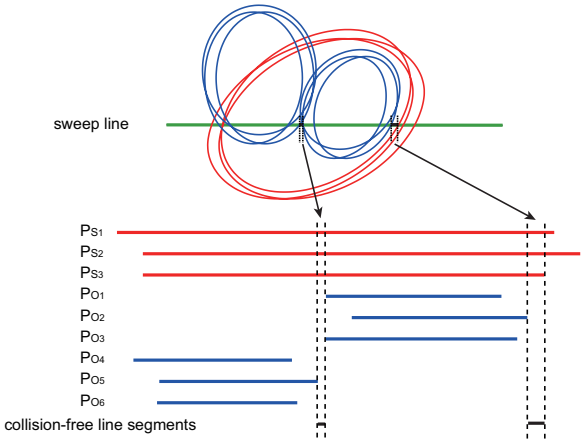

Fig. 2. The detection scheme for the collision-free line segments on each sweep line using the logic in Eq. 6.

surface is then stretched using the inverse of the previous affine transformation.

The exact boundary of $E_1 \oplus E_2$ can be finally represented in closed form as

$$\mathbf{x}_{eb} = R_2 \Delta(\mathbf{a}_2/r) R_2^T R_1' \mathbf{x}_{ofs}. \tag{4}$$

where

$$\begin{aligned} \mathbf{x}_{ofs}(\phi) &= \mathbf{x}(\phi) + r\,\mathbf{n}(\phi), \\ \mathbf{n}(\phi) &= \frac{\nabla\Phi(\mathbf{x}(\phi))}{\|\nabla\Phi(\mathbf{x}(\phi))\|}, \\ \nabla\Phi(\mathbf{x}) &= 2\Lambda^{-2}(\mathbf{a}_1)\,\mathbf{x}. \end{aligned} \tag{5}$$

$\mathbf{x}_{ofs}(\phi)$ is a parameterized offset hyper-surface of an orientable, closed, and differentiable hyper-surface $\mathbf{x}(\phi) \in \mathbb{R}^n$ with the offset radius r is defined. In this case, $\mathbf{x}(\phi)$ is the explicit expression of the shrunk version of E_1, and R_1' describes the orientation of the same ellipsoid. $\mathbf{n}$ is the outward-pointing unit surface normal. and $R_2 \in SO(n)$ describes the orientation of E_2. Similarly, the Minkowski difference of two ellipsoids, i.e., $E_1 \ominus E_2 = \{\mathbf{y} | \mathbf{y} + \mathbf{x}_2 \in E_1, \mathbf{x}_2 \in E_2\}$, can be calculated in closed-form too. The only difference is that in (5) the offset distance r is changed to $-r$.

III. Characterization of the Collision-Free C-Space with Robot, Obstacle(s) and Environment Represented as Finite Unions of Ellipsoids

In Sec. II, the parametric equations of the collision-free boundaries of ellipsoid-ellipsoid interactions in two different types were introduced.

In this section, we first present a detection scheme for the collision-free spaces of finite unions of ellipsoids, given the complete description of the collision-free space boundaries of each ellipsoid-ellipsoid interaction. We use a simple planar example to illustrate this detection scheme. In our example, the robot is the union of three ellipses (which we call a "rabbit" with a face and two ears). The scenario is that this rabbit is allowed to roam inside an elliptical environment cluttered by large elliptical obstacles (see Fig. 1 (a)). The boundaries of the collision-free space for the robot-obstacle interaction

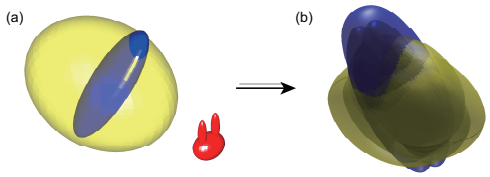

Fig. 3. (a) The example of one rabbit-shaped robot translating inside an ellipsoidal arena that contains an ellipsoidal obstacle. (b) The overlaid boundaries of the collision-free space for the robot-obstacle interaction (blue-shaded surfaces) and those for the robot-environment interaction (yellow-shaded surfaces) for the rabbit face, ear 1 and ear 2.

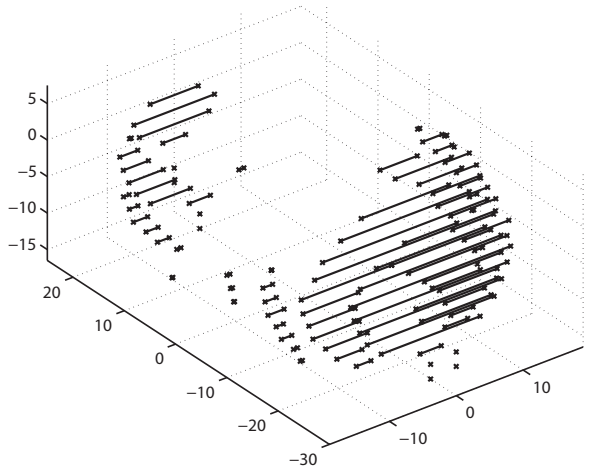

Fig. 4. The collision-free space using the detection scheme for the example in Fig. 3.

(blue) and those for the robot-environment interaction (red) are shown in Fig. 1 (b). After overlaying all the boundaries together, the collision-free space (the green-shaded region) are the regions which are inside all the red curves and outside all the blue curves.

The detection scheme can be applied to more general cases as long as robot(s), obstacle(s) and environment(s) are constructed by finite unions of ellipses. Suppose that the shape of robot is a combination of k ellipses named $E_1, E_2, \cdots, E_k$ and $g_2, g_3, \cdots, g_k$ represent the rigid body motions between the first ellipse E_1 and the other ellipses $E_2, E_3, \cdots, E_k$. Each rigid-body motion consists of a rotation-translation pair $g_i = (R_i, \mathbf{t}_i)$. Let the collision-free space for $E_1, E_2, \cdots, E_k$ be $C_1, C_2, \cdots, C_k$, and then the collision-free space of the whole robot can be characterized as $C_1 \cap (g_2 \circ C_2) \cap (g_3 \circ C_3) \cap \cdots \cap (g_k \circ C_k)$ where $g_i \circ C_j \doteq R_i C_j + t_i$.

To detect these regions, we generate a set of sweep lines parallel to the y-axis (in general, it can be a set of parallel lines along any direction). For each sweep line, its intersection points with all the curves are detected and saved in pairs, or intervals. Here, let the intersecting line segments between the sweep line and a red curve (robot-environment interaction boundary) be P_{S_i}, and those between the sweep line and a blue curve (robot-obstacle interaction boundary) be P_{O_i}. Then the collision-free line segments on each line (P_{CF}) can be

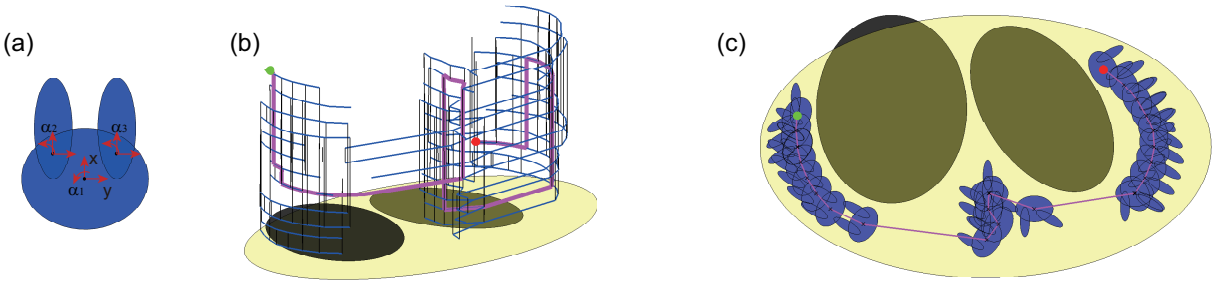

Fig. 5. (a) the rabbit-shaped robot with 5 dofs — the translation of the rabbit (in x and y directions) and rotations of the rabbit face and two ears (α_1, α_2 and α_3), where $0 \le \alpha_1 \le 2\pi$, $0 \le \alpha_2 \le \pi/2$, and $-\pi/2 \le \alpha_3 \le 0$. (b) An example of the highway roadmap for the robot with 5 dofs. Each layer of the roadmap representing a different combination of α_1, α_2 and α_3. The layers of the roadmap are interconnected at the nodes which are relatively close to one other (based on the Euclidean distance). A path is found using Dijkstra's algorithm given the start and goal (shown as magenta lines). (c) The steps of the motion of the robot overlaid onto the shortest path.

represented as

$$P_{CF} = P_{S_1} \cap P_{S_2} \cap P_{S_{n_s \cdot k}} - P_{O_1} \cup P_{O_2} \cup P_{O_{n_o \cdot k}}, \qquad (6)$$

where n_s and n_o are the numbers of ellipses that are used to construct the environment and the obstacles, respectively. This simple logic for the detection scheme of the collision-free line segments is illustrated in Fig. 2.

For the 3D cases, robot(s), obstacle(s) and environment(s) are constructed by finite unions of ellipsoids. To detect the collision-free space of the whole robot for both robot-environment and robot-obstacle interactions, we first slice all the collision-free boundary surfaces along the z-axis. By transforming 3D surfaces to a set of 2D curves with respect to each z-coordinates, we are able to apply the detection scheme for the 2D elliptical case. A 3D example is shown in Fig. 3 and the resulting collision-free space using the detection scheme is shown in Fig. 4.

We note that we can also slice along the x-axis or the y-axis and sweep scan along one of the rest two axes as well. Or we can combine the sampling schemes in different directions together to obtain a richer sampling space. However, in our highway roadmap planner which is based on cell decompositions (see Sec. IV), sampling resolution is not very critical, especially for smooth surfaces in our cases. Fig. 4 illustrates the collision-free space for the same example in Fig. 3. The collision-free space is constructed by stacking all the collision-free regions on each thin slice together along the z-axis.

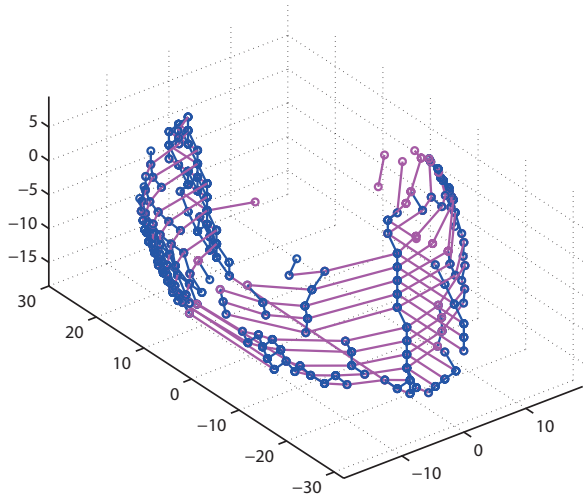

Fig. 6. The 3D highway system for the example in Fig. 3.

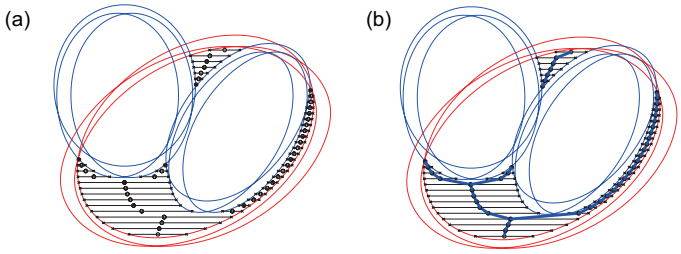

Fig. 7. An example of generating the highway roadmap for the rabbit-shaped robot with 2 dofs (only translations in x and y directions). (a) The collision-free line segments and the midpoints on these line segments along each sweep line, (b) the adjacency graph connecting these midpoints, i.e., the highway roadmap, (shown as blue lines).

IV. Path Planning Approach: Highway Roadmap Planner

A. Constructing highway roadmaps

To build a highway roadmap in 3D, we first construct a planar highway roadmap in x-y plane, and then stack and connect all the planar roadmaps together along the z-axis. Here, we call it a "highway" roadmap since in the analogy, this roadmap provides the routes that are furthest away from the local obstacles. With the knowledge of collision-free line segments along each sweep line, the collision-free space can be naturally approximated by a union of trapezoidal cells, with the two parallel sides of each trapezoidal cell as two neighboring collision-free line segments in the vertical direction. Two cells are adjacent if they share a common boundary, i.e., a common horizontal extension. Since a trapezoid is a convex set, any two points on the boundary of a trapezoidal cell can be connected by a straight line segment that does not intersect any obstacle. Therefore in our planner roadmap, we can connect the midpoints of the horizontal extensions of the

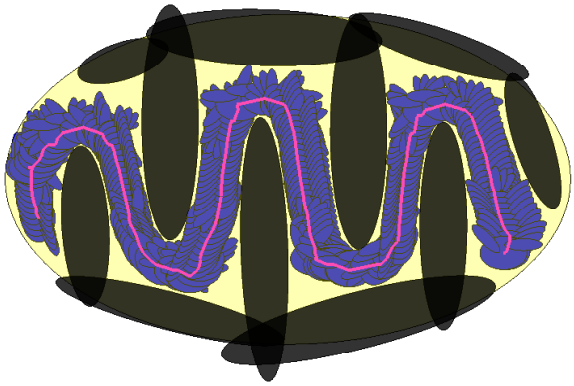

Fig. 8. A planar example with the narrow passage problem. We note that the rabbit's two ears cannot fold back inward to the face since the rabbit model has the range of motion on the ears, i.e., $0 \leq \alpha_2 \leq \pi/2$, and $-\pi/2 \leq \alpha_3 \leq 0$.

adjacent trapezoidal cells, one at each time, and the resulting adjacency matrix of these midpoints could serve as a roadmap of the collision-free regions.

B. Path planning examples

In the planar case, when the robot only has 2 degrees of freedom, the translations in x and y directions, with the face and ear angles fixed. For the two dimensional case, the Euclidean distance is used as the metric. Fig. 7 gives an example of how a planar roadmap is connected on a x -y plane. When the robot has more dofs, a higher dimensional roadmap is needed. We also construct the rabbit-shaped robot with 5 dofs, the translations in x and y directions, and the rotations of the rabbit face and two ears, α_1, α_2 and α_3, respectively (see Fig. 5 (a)). In Fig. 5 (b), each layer of the roadmap representing a different combination of α_1, α_2 and α_3. The layers of the roadmap can be interconnected at the nodes which are relatively close to one other (based on the Euclidean distance). In this case, the metric used in the graph search is $|\Delta P| + w_1|\Delta\alpha_1| + w_2|\Delta\alpha_2| + w_3|\Delta\alpha_3|$, where $|\Delta P|$ is the Euclidean distance between the centers of the faces and $|\Delta\alpha_1|\cdots|\Delta\alpha_3|$ and $w_1 \cdots w_3$ are the absolute differences and the corresponding weights of the face and two ear angles, respectively. We note that self-collision between the different parts of the robot can be eliminated in advance, and we focus only on robot-obstacle and robot-environment interactions. A pseudo code can be found in Alg. 1.

Like other methods that involve cell decompositions, the path planning is usually done in two steps — first, the planner determines the cells that contain the start and goal, respectively, and then the planner searches for a path in the adjacency graph. In our planner, we use the Dijkstra's algorithm [29] to search for the shortest path.

With our closed-form characterizations of the collision-free C-space along with the highway roadmap, the collision checking is almost unnecessary. Vast volumes within C-space that correspond to robot/obstacle or robot/environmental-boundary collisions can be thrown away directly. When using standard methods for motion planning, collision checking can be computational intensive, especially when the ratio of the volumes between the collision-free C-space and the whole C-space is small. Our complete characterization of the collision-free C-space becomes particularly useful for the "narrow passage" sampling problem. Also, with this highway roadmap system, instead of evaluating the edges between nodes every time, once the start and the goal are connected to the roadmap, a path is instantly constructed. For simplicity, we design the collision checking algorithm between ellipsoids by first uniformly sampling points on the surfaces (curves for ellipse) of the any two ellipsoids so that there are 50 sampled vertices on an ellipse and 360 vertices on an ellipsoid. For two ellipsoids, we substitute the coordinate $\mathbf{x} = (x,y,z)^T$ of the vertices of one ellipsoid into the implicit equation of the other, and report collision when $\Phi(\mathbf{x}) > 1$. In the later section, we will refer this simple approach as the "traditional collision checking" method. Though it is not the optimal way to check collision between ellipsoids, the significantly better performances of the proposed methods still manifest the advantages of our characterization of the collision-free C-spaces. The implementation of ellipsoid specific collision checking algorithm such as [22] will be left to future work.

Algorithm 1 Highway Roadmap Algorithm

procedure SAMPLING AND LAYER CONNECTION
Given configuration P and ellipsoidal bodies E
  $P_{start}, P_{goal}, E_{arena}, E_{obs}, E_{robot}(\alpha_1, \alpha_2, \alpha_3)$, where $\mathbb{A} = \{\alpha_1, \alpha_2, \alpha_3\}$ represent the discretized angles as shown in Fig. 5 with sizes of n_1, n_2 and n_3 respectively.
for: $i \leftarrow 1, \ldots, n_1 \times n_2 \times n_3$
  $Bound_{Mink} \leftarrow \text{Minkowski}\,(E_{arena}, E_{obs}, E_{robot}(\mathbb{A}(i)))$
  $P_{mid} \leftarrow \text{Highway}(Bound_{mink})$
  $M_{adj} \leftarrow \text{BuildAdjacencyMatrixWithinLayer}\,(P_{mid})$
  $M_A \leftarrow M_A + M_{adj}$ //Form the complete adjacency matrix
$M_A \leftarrow \text{ConnectSubadjacencyMatrixBetweenLayers}\,(M_A)$
$Path \leftarrow Dijkastra(M_A, P_{start}, P_{goal})$

1) Planar examples: In this section, a path planning problem in the planar case is given. The robot, obstacle(s) and environment are constructed by unions of ellipses (see Figs. 5 (c) and 8). We compare our approach with two popular sampling-based path planning algorithms, a probabilistic roadmap [1] and a standard rapidly-exploring random tree (RRT) [2]. In our examples, we compare the computational speeds using the RRT and the PRM with our approach for 2 different scenarios — 1) when the volume of the collision-free C-space is relatively large compared to that of the whole C-space and 2) when the volume of the collision-free C-space is relatively small and a narrow passage problem arises (Fig. 8). The codes are all written using MATLAB 2015b and run on a Lenovo ThinkCentre M83 with Intel(R) Core(TM) i7-4790 CPU@3.60GHz Processor and 12 GB RAM.

In the experiments, 5 trials are generated using both the RRT and the PRM with the computational speeds shown in Tabs. I and II. In our approach, the collision-free C-space and the highway roadmap system only need to be constructed once. In the first scenario (see Fig. 5), when the robot has a large free motion space, less time is needed to find a feasible path using the RRT and the PRM, with the average speed of 2.5 seconds and 5.02 seconds, respectively, compared to 3.3 seconds using our approach. But as the free motion space of the robot significantly shrunk (as shown in Fig. 8), with our approach, it only takes 37.3 seconds to compute a feasible path, compared to an average speed of 1317 seconds using the RRT and 1346 seconds using the PRM.

TABLE I
THE TIME SPENT ON FINDING A PATH USING THE RRT BASED ON 5 TRIALS. EX. 1 AND EX. 2 ARE THE EXAMPLES SHOWN IN FIGS. 5 AND 8, RESPECTIVELY. THE NUMBERS OF NODES USED IN EACH TRIAL TO FIND A FEASIBLE PATH BY THE RRT ARE ALSO SHOWN IN THE TABLE.

RRT trial	1	2	3	4	5	Ave	Ours
Ex. 1 time (s)	2.9	1.8	3.6	1.2	3.0	2.5	3.3
Ex.2 time (s)	1325	869	1616	824	1953	1317	53
# of nodes	3310	3046	4918	1913	5486	3734	

TABLE II
THE TIME SPENT ON FINDING A PATH USING THE PRM BASED ON 5 TRIALS. EX. 1 AND EX. 2 ARE THE EXAMPLES SHOWN IN FIGS. 5 AND 8, RESPECTIVELY.

PRM trial	1	2	3	4	5	Ave	Ours
Ex.1 time (s)	5.9	4.8	4.6	5.9	3.9	5.0	3.3
Ex.2 time (s)	1138	1117	2243	1257	977	1346	53

V. CONCLUSION

In this paper, we present an approach to parameterizing the exact boundaries of the Minkowski sum and difference of two ellipsoids. Based on this closed-form representation, we present a new method for parameterizing the collision-free regions of the C-space in robot motion planning. The robot(s), obstacle(s) and the environment(s) can be described by ellipses/ellipsoids or finite unions of ellipses/ellipsoids. The parametric representations of the collision-free regions in C-space are given. With our closed-form characterizations of the free space, for single-rigid-body, a highway roadmap system is constructed to connect the collision-free C-spaces. More sophisticated ellipsoid-oriented collision checking algorithms will be implemented to compare the planners' performances in a more comprehensive way.

REFERENCES

[1] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[2] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," *In*, vol. 129, no. 98-11, pp. 98–11, 1998.

[3] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *Access, IEEE*, vol. 2, pp. 56–77, 2014.

[4] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 673–692, 2004.

[5] G. Varadhan and D. Manocha, "Star-shaped roadmaps-a deterministic sampling approach for complete motion planning.," in *Robotics: Science and Systems*, vol. 173, Citeseer, 2005.

[6] L. Janson, B. Ichter, and M. Pavone, "Deterministic sampling-based motion planning: Optimality, complexity, and performance," *arXiv preprint arXiv:1505.00023*, 2015.

[7] S. Hirsch and D. Halperin, "Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane," in *Algorithmic Foundations of Robotics V*, pp. 239–255, Springer, 2004.

[8] J.-M. Lien, "Hybrid motion planning using minkowski sums," *Proceedings of Robotics: Science and Systems IV*, 2008.

[9] J.-M. Lien, "Point-based minkowski sum boundary," in *Pacific Conference on Computer Graphics and Applications*, pp. 261–270, IEEE, 2007.

[10] J. Pan and D. Manocha, "Efficient configuration space construction and optimization for motion planning," *Engineering*, vol. 1, no. 1, pp. 46–57, 2015.

[11] D. Halperin, J.-C. Latombe, and R. H. Wilson, "A general framework for assembly planning: The motion space approach," *Algorithmica*, vol. 26, no. 3-4, pp. 577–601, 2000.

[12] E. Hartquist, J. Menon, K. Suresh, H. Voelcker, and J. Zagajac, "A computing strategy for applications involving offsets, sweeps, and Minkowski operations," *Computer-Aided Design*, vol. 31, pp. 175–183, Mar. 1999.

[13] E. Behar and J.-M. Lien, "Dynamic Minkowski sum of convex shapes," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3463–3468, IEEE, 2011.

[14] L. Guibas, L. Ramshaw, and J. Stolfi, "A kinetic framework for computational geometry," in *Proc. 24th Annu. IEEE Sympos. Found. Comput. Sci*, pp. 100–111, 1983.

[15] E. Fogel and D. Halperin, "Exact and efficient construction of Minkowski sums of convex polyhedra with applications," *Computer-Aided Design*, vol. 39, pp. 929–940, Nov. 2007.

[16] P. Hachenberger, "Exact Minkowski Sums of Polyhedra and Exact and Efficient Decomposition of Polyhedra into Convex Pieces," *Algorithmica*, vol. 55, pp. 329–345, Aug. 2008.

[17] P. K. Agarwal, E. Flato, and D. Halperin, "Polygon decomposition for efficient construction of Minkowski sums," *Computational Geometry*, vol. 21, no. 1-2, pp. 39–61, 2002.

[18] Y. Yan and G. S. Chirikjian, "Closed-form characterization of the minkowski sum and difference of two ellipsoids," *Geometriae Dedicata*, vol. 177, no. 1, pp. 103–128, 2015.

[19] P. Goodey and W. Weil, "Intersection bodies and ellipsoids," *Mathematika*, vol. 42, pp. 295–304, Dec. 1995.

[20] J. W. Perram and M. Wertheim, "Statistical mechanics of hard ellipsoids. I. Overlap algorithm and the contact function," *Journal of Computational Physics*, vol. 58, pp. 409–416, May 1985.

[21] K. Chan, W. W. Hager, S.-J. Huang, P. M. Pardalos, O. A. Prokopyev, and P. Pardalos, *Multiscale Optimization Methods and Applications*, vol. 82 of *Nonconvex Optimization and Its Applications*. Boston: Kluwer Academic Publishers, 2006.

[22] Y.-K. Choi, J.-W. Chang, W. Wang, M.-S. Kim, and G. Elber, "Continuous collision detection for ellipsoids," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 2, pp. 311–325, 2009.

[23] C. Dube, M. Tsoeu, and J. Tapson, "A model of the humanoid body for self collision detection based on elliptical capsules," in *International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2397–2402, IEEE, 2011.

[24] A. Best, S. Narang, and D. Manocha, "Real-time reciprocal collision avoidance with elliptical agents," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2016.

[25] L. Kavraki, "Computation of configuration-space obstacles using the fast Fourier transform," *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 408–413, June 1995.

[26] A. B. Kyatkin and G. S. Chirikjian, "Computation of Robot Configuration and Workspaces via the Fourier Transform on the Discrete-Motion Group," *The International Journal of Robotics Research*, vol. 18, pp. 601–615, June 1999.

[27] Y. Wang and G. S. Chirikjian, "A new potential field method for robot path planning," in *IEEE International Conference on Robotics and Automation. Symposia Proceedings*, vol. 2, pp. 977–982, IEEE, 2000.

[28] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke, *Stochastic geometry and its applications*. John Wiley & Sons, 2013.

[29] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.