

**Geometric Motion Planning Methods for Robotics and Biological
Crystallography**

by

Yan Yan

A dissertation submitted to The Johns Hopkins University in conformity with the
requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

February, 2014

© Yan Yan 2014

All rights reserved

Abstract

This dissertation presents several novel approaches to motion planning problems by incorporating useful geometrical models. Two different approaches are addressed in the contexts of biological crystallography and robot motion planning, respectively.

In the context of biological crystallography, the method of molecular replacement (MR) is frequently used to obtain the phase information for a crystallographic unit cell packed with copies of a macromolecule of unknown conformation. This is important because an X-ray diffraction experiment on its own does not provide full structural information.

MR works quite well for single-domain proteins that can be treated as single rigid-bodies. However, for multi-domain structures and complexes, computational requirements can become prohibitive due to the “curse of dimensionality”. We introduce a new phasing approach by using geometric constraints on protein crystals to generate packing models and therefore replace the computationally expensive MR searches in a continuous configuration space with a search on a relatively small discrete set of candidate packing arrangements.

In the context of robot motion planning, an approach based on the closed-form representation of the robot’s collision-free configuration space (C-space) is presented. In this

ABSTRACT

approach, the robot, obstacles, and environment are modeled as ellipsoids or finite unions of ellipsoids in the workspace. These objects can be quite general, approximated by unions of different ellipsoidal bounding boxes, and this approach represents an alternative to polyhedral representations of bodies. The approach builds on the core idea that the space of collision-free motions of one moving ellipsoid relative to one fixed ellipsoid can be characterized exactly in closed form. This means that for ellipsoidal bodies, there is no need to sample and discard configuration suspected of being in collision. This approach of “knowing where to look” for path planning in C-spaces is especially useful for the narrow passage problem.

Additionally, the quality of samples on the rotation and motion spaces is important in these applications, especially for sampling based motion planning methods. In this dissertation, we also developed a group-theoretic based approach to achieve almost-uniform sampling of these spaces and to further facilitate a broader class of motion planning problems. This approach is based on decomposing the rotation and Special Euclidean groups into identical Voronoi cells centered on the elements of their discrete subgroups. Within each cell, regular grids in exponential coordinates are used to achieve almost-uniform sampling at any level of resolution, without having to store large numbers of coordinates, and without requiring sophisticated data structures. We also analyze the shape of these cells, and discuss how this new method can be used in the context of conformational searches in the field of structural biology and robotics contexts.

ABSTRACT

Thesis Adviser:

Dr. Gregory S. Chirikjian, Professor, Johns Hopkins University

Thesis Committee:

Dr. Gregory S. Chirikjian, Professor, Johns Hopkins University

Dr. Louis L. Whitcomb, Professor, Johns Hopkins University

Dr. Marin Kobilarov, Assistant Professor, Johns Hopkins University

Acknowledgments

The past four and half years provides me tremendous professional and person growth. Thanks to this journey, now I have a model of the world and myself much better than before.

First of all, I'd like to thank my adviser Dr. Gregory Chirikjian for his guidance and support during my studies. His dedication to research has inspired me to work harder and keep improving. I am grateful for the time, attention and advice he has generously given.

I would like to thank my committee members, Dr. Louis Whitcomb and Dr. Marin Kobilarov for their insightful comments regarding this thesis and defense and for their support and enthusiasm towards the problems addressed in this work.

The quality of a graduate school experience is tied with the people that surround you. I have enjoyed spending my time with many friends from the LCSR lab and from the Mechanical Engineering department. Special thanks go (in no particular order) to Yixin, Topher, Xingchi, Paul, Sue Sue, Hui, Ying, Xi, Yi, Wooram, Matt, Kevin, Kendal, Josh, Qianli, Taosha, Xiao, Manu, Eatai, John, Shahin, Ehsan. My work life would have been really lonely without you.

ACKNOWLEDGMENTS

Last, and by no means least, I would like to thank my parents. Till now, I have already been studying abroad for six and half years and I haven't been home for the past two years. I know it could be very hard on my parents, especially my mom, the best girlfriend in my life. But they always give me tremendous support for my work even if they miss me so much. Also, I thank them for the best childhood, which grew the happiness at my core and for always being proud of me, which gives me strength to conquer occasional self-doubt and different kinds of challenges in life.

Dedication

To my parents

Contents

Abstract	ii
Acknowledgments	v
List of Tables	xiii
List of Figures	xiv
1 Introduction	1
1.1 Overview	1
1.2 Review of Motion Planning Methods	4
1.3 Contributions	10
2 Mathematical Overview	13
2.1 Basic Concepts in Group Theory	13
2.2 Rigid-body Rotation and Motion Spaces	20
2.3 Crystallographic Symmetry	24

CONTENTS

2.4	Voronoi Diagrams	27
3	A Phasing Approach Using Packing Models for Multi-Domain Structures in Crystals	30
3.1	Introduction	32
3.2	Essentials of Macromolecular X-Ray Crystallography (MX)	35
3.3	The Multi-Domain Molecular Replacement Method (MMR)	38
3.4	Phasing by Packing	40
3.4.1	Collision detection for generating candidate packing arrangements	42
3.4.2	A Gaussian-based potential function for generating candidate packing arrangements	44
3.5	Numerical Results	48
3.5.1	A planar example: $p1$ symmetry	50
3.5.2	Spatial examples: $P1$ and $P2_12_12_1$ symmetries	53
3.5.2.1	Results in $P1$ symmetry	57
3.5.2.2	Results in $P2_12_12_1$ symmetry	60
3.6	Conclusion	68
4	An Almost-Uniform Sampling of Rigid-Body Rotations and Motions	71
4.1	Introduction	72
4.2	Definition and Terminology	77
4.3	An Almost-Uniform Sampling Approach for $SO(3)$	79

CONTENTS

4.3.0.3	Discrete subgroups of $SO(3)$	79
4.3.0.4	Generating Voronoi cells of $SO(3)$	80
4.3.0.5	Curvature of the boundaries of the Voronoi cells for $SO(3)$	85
4.3.0.6	Distortion Measure of the samplings for $SO(3)$	87
4.3.1	Application in the context of a crystal packing problem	90
4.3.2	Improved almost-uniform sampling on $SO(3)$ based on double- coset decompositions	92
4.4	An Instructive Approach for Almost-Uniform Sampling for $SE(2)$	99
4.5	Conclusion	103
5	Closed-Form Characterization of Minkowski Sum and Difference of Two El- lipsoids	105
5.1	Introduction	106
5.2	Closed-Form Characterization of Minkowski Operations of Two Ellipsoids .	110
5.2.1	The Minkowski sum of two ellipsoids	110
5.2.2	The Minkowski difference of two ellipsoids	114
5.3	Approximation of the Minkowski Operations of Two Ellipsoids	116
5.3.1	Ellipsoidal approximation of the offset surfaces	116
5.3.2	Ellipsoidal approximation based on Gaussian functions	119
5.4	The Volume of the Minkowski Sum of Two Ellipsoids	121
5.4.1	Bounds on the Minkowski sum volume of two ellipsoids via Steiner's formula (for planar cases)	124

CONTENTS

5.4.2	Bounds on the Minkowski sum volume of two ellipsoids via Steiner's formula (for spatial cases)	126
5.4.2.1	Bounds on the total mean curvature of a triaxial ellipsoid	128
5.4.2.2	Bounds on the surface area of a triaxial ellipsoid	137
5.4.3	Numerical comparison of different bounds on the Minkowski sum volume of two ellipsoids	141
5.4.3.1	The results for the planar case	141
5.4.3.2	The results for the spatial case	144
5.4.4	Volume estimates on the Minkowski sum volume of two ellipsoids derived from the bounds	145
5.5	A Kinematic Formula for Containment	150
5.5.1	The Formula for planar cases	150
5.5.2	The Formula for spatial cases	153
5.6	Conclusion	155
6	Path Planning Based on Characterization of Collision-Free Configuration-Space for Ellipsoidal Bodies, Obstacles, and Environment	157
6.1	Introduction	159
6.2	Characterization of the Collision-free C-Space with Robot, Obstacle(s) and Environment Represented as Finite Unions of Ellipsoids	161
6.3	Path Planning Approach 1: Highway Roadmap Planner	166
6.3.1	Constructing highway roadmaps	166

CONTENTS

6.3.2	Algorithm for removing redundant nodes on the Highway roadmap	170
6.3.3	Path planning examples	172
6.3.3.1	Planar examples	173
6.3.3.2	Spatial examples	175
6.4	Path Planning Approach 2: Sampling-Based Approach with Safety Function for Articulated Bodies	176
6.5	Conclusion	182
7	Conclusion	186
A	Matlab code for space group calculations	190
A.1	Matlab Code for Normal Subgroup Test	190
A.2	Matlab Code for the Graphical User Interface (GUI) of Group Calculations	196
	Bibliography	209
	Vita	235

List of Tables

3.1	The dimensions and ranges of motion of the rabbit packing model.	55
3.2	10 Numerical trials for the planar crystal packing example. The angles are all in degrees.	55
3.3	The dimensions and ranges of motion of the packing model in $P1$ symmetry.	60
3.4	The motions of the target structure and the set of candidates in the example of Fig. 3.13 in $P2_12_12_1$ symmetry. The angles are all in degrees.	63
3.5	The dimensions and ranges of motion of the packing model in $P2_12_12_1$ symmetry.	65
3.6	The motions of the target structure and the set of candidates in the example of Fig. 3.18 in $P2_12_12_1$ symmetry. The angles are all in degrees.	67
6.1	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. 6.7 and 6.11, respectively.	174
6.2	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. 6.7 and 6.11, respectively. The numbers of nodes used in each trial to find a feasible path by the PRM are also shown in the table.	175

List of Figures

2.1	The five chiral wallpaper groups: (a) $p1$; (b) $p2$; (c) $p4$; (d) $p3$; (e) $p6$	17
2.2	The five Platonic solids: (a) tetrahedron; (b) cube; (c) octahedron; (d) icosahedron (e) dodecahedron.	18
2.3	Objects arranged with $P2_12_12_1$ symmetry. (a) 3D view; (b)-(d) projections.	27
2.4	Coordinated movement of these objects that maintains $P2_12_12_1$ symmetry. (a) 3D view; (b)-(d) projections.	28
3.1	The schematic diagram of a macromolecular X-ray crystallography experiment. Only the diffraction data can be obtained (on the film) and the phase information of the molecule is killed.	33
3.2	The rabbit packing model motivated by the 3-domain protein Lactoferrin.	43
3.3	Flowchart of our approach to finding candidate phasing models. The candidate packing arrangements are generated by collision detections.	45
3.4	The comparison between a single Gaussian with a mixture of Gaussians.	47
3.5	Flowchart of our approach to finding candidate phasing models. The candidate packing arrangements are generated by minimizing the Gaussian-based potential function.	49
3.6	The size of the non-collision candidate set with different m_b values under 3 different definition grids (in 30-, 40- and 60-degree increments, respectively). $ S_{\text{cand.}}(m_b) $ denotes the number of non-collision candidates in this set.	52
3.7	The comparison of the rabbit shape with the contours of the Gaussian mixture function ($m_b = 0.2$).	53
3.8	The comparison of (a) collision checking results with (b) GCF values (with $m_b = 0.2$) in the θ_1 - θ_2 plane (with $\theta_3 = -90$ degrees). In (a), black pixels represent the collision-free configuration and white ones are in collision. In (b), the pixels with darker colors represent the configuration with lower GCF values, and vice versa.	54
3.9	3 candidate packing arrangements for the planar crystal packing example.	56
3.10	The planar crystal packing example with the target structure randomly sampled in the space.	56

LIST OF FIGURES

3.11	(a) Illustration of 3 degrees of freedom in the packing model. Examples of packing configuration (b) without collisions (c) with collisions (indicated in yellow).	58
3.12	An example of packing results with the target structure on the grid in $P1$ symmetry.	61
3.13	An example of packing results with the target structure randomly sampled in the space in $P1$ symmetry.	61
3.14	The trend of errors before and after applying the RRT in $P1$ symmetry.	62
3.15	Illustration of 10 degrees of freedom in the packing model.	62
3.16	The examples of packing configuration (a) with collisions and (b) without collisions.	64
3.17	An example of packing results with the target structure on the grid in $P2_12_12_1$ symmetry.	68
3.18	An example of packing results with the target structure randomly sampled in the space in $P2_12_12_1$ symmetry.	69
3.19	The trend of errors before and after applying the RRT in $P2_12_12_1$ symmetry.	70
4.1	(a) Voronoi cells of the octahedral group in $SO(3)$ are represented in exponential coordinates. Each color represents a Voronoi cell in $SO(3)$. We note that because the exterior Voronoi cells cover the interior ones in the figure the regions with different colors may appear to be patches on a sphere but actually represent for sites in a solid ball. (b) The center Voronoi cell that corresponds to the identity matrix is represented in exponential coordinates. The yellow-shaded ball represents the solid ball in \mathbb{R}^3 with radius π	83
4.2	The center Voronoi cell shifted by a rotation matrix R in the icosahedral group VS the Voronoi cell that corresponds to R . The complete overlapping between the two sites shows that the contents of the Voronoi cell on R can be replicated by the contents of the Voronoi cell on the identity by shifting the contents by R	83
4.3	Polyhedra in $\mathbb{R}^3 \cong so(3)$ depicting the center Voronoi cells in $SO(3)$ corresponding to the (a1) tetrahedral (b1) octahedral and (c1) icosahedral groups. The wire frame plots of these polyhedra with an example of uniform grids in \mathbb{R}^3 filled in for the (a2) tetrahedral, (b2) octahedral and (c2) icosahedral groups.	86

LIST OF FIGURES

4.4 The distortion of the samples on $SO(3)$ using (a) our sampling method and (b) ZXZ Euler angles. In our sampling method, all the three discrete subgroups are evaluated on the norm of the Cartesian coordinates under exponential parameterization, $\|\mathbf{x}\|$. The maximum distortion for each discrete subgroup is marked in the figure. The icosahedron group should be used since it has the smallest Voronoi cell and results in the smallest distortion. By using the ZXZ Euler angles, the two Z rotations α and γ are sampled uniformly and the middle Euler angle β is sampled based on $\beta = \cos^{-1}(t)$, where t is sampled uniformly. In this case, the distortion is evaluated on each value of β from 0 to π 89

4.5 (a) Illustration of 3 degrees of freedom in the packing model. Examples of packing configuration (b) without collisions (c) with collisions (indicated in yellow) (the same model is used in Chap. 3). 93

4.6 The number of collision-free conformations VS the number of sample points on $SO(3)$ generated by (a) our sampling method, (b) sampling of Euler angles. 94

4.7 The center Voronoi cell in a single-coset-space decomposition ($F_{K \setminus SO(3)} = F_{SO(3)/K}$ = yellow-shaded region) with K as the icosahedral group, and the center Voronoi cells in double-coset-space decompositions (red-shaded regions) with K as the icosahedral group for all cases and H as the conjugated tetrahedral group (a), the conjugated octahedral group (b) and the conjugated icosahedral group (c), respectively. 96

4.8 The sliced plots of the union of the Voronoi cells from the double-coset space $H \setminus G / K$, where H is the icosahedral group, K is a conjugated version of the icosahedral group and G is $SO(3)$. Here $K = g \circ H \circ g^{-1}$, where $g \in SO(3)$, and $H \cap K = \{e\}$ 97

4.9 The maximum distortions from different double and single coset spaces. . . 98

4.10 The dodecahedral cell (yellow-shaded region) and the tetrahedral wedge (red-shaded region). The dodecahedral cell can be decomposed into 60 identical tetrahedral wedges like this, with 5 packed to form a pyramid with pentagonal base corresponding to a face of the dodecahedron. 99

4.11 (a) $SE(2)$ illustrated in \mathbb{R}^3 and the parallelogrammatic lattice for $p1$; (b) the Voronoi cells for $SE(2)$ based on $p1$ 101

4.12 $F_{\Gamma_i \setminus SE(2)}$ for 5 chiral wallpaper groups $p1$ (a), $p2$ (b), $p4$ (c), $p3$ (d) and $p6$ (e). 102

LIST OF FIGURES

5.1 The algorithm for obtaining the parametric representation of the boundary of the Minkowski sum of two ellipses E_1 and E_2 . In (a), both ellipses are shrunk in the direction with the rotation angle θ_2 (see (5.3)) until E_2 becomes a circle. After the shrinking process, E_1 remains elliptical but with different semi-axis lengths and rotation angle, whereas E_2 becomes a circle (see (b)). The Minkowski sum in this transformed space is then an offset surface. In (c), everything is stretched back in the same direction until the surrounding circle becomes the original version of E_2 again. The shaded region in (c) represents $E_1 \oplus E_2$. We note that the shaded regions in (b) and (c) that may appear to be elliptical are actually not ellipses, and the amount that they deviate from being ellipses depends on the eccentricity of the elliptical bodies. 115

5.2 The algorithm for obtaining the parametric representation of the boundary of the Minkowski difference of two ellipses E_1 and E_2 . This presumes that the radius of curvature of the resulting circle/sphere is smaller than the smallest semi-axis length of the bounding shrunk ellipsoid. The algorithm follows the same steps as Fig. 5.1. The shaded region in (c) represents $E_1 \ominus E_2$ 117

5.3 An planar example: the exact boundary (black), approximated elliptical boundary 1, based on the approximation of the offset curves (blue) and approximated elliptical boundary 2, based on Gaussian functions (red) of the Minkowski sum of two ellipses. The semi-axis lengths of E_1 and E_2 are 6, 2, and 10, 2, respectively. The six subplots correspond to six different angles (θ_2) of E_2 122

5.4 A 3D ellipsoidal example: the exact boundary (green), approximated ellipsoidal boundary 1, based on the approximation of the offset surfaces (blue) and approximated ellipsoidal boundary 2, based on Gaussian functions (red) of the Minkowski sum of two ellipsoids. The semi-axis lengths of E_1 and E_2 are 20, 30, 40, and 6, 2, 2, respectively. The ZXZ Euler angles of E_1 and E_2 are $\alpha_1 = 0, \beta_1 = 0, \gamma_1 = 0$ and $\alpha_2 = 0, \beta_2 = 0, \gamma_2 = 0$, respectively. 123

5.5 Upper and lower bounds, $M_{ub1}, M_{ub2}, M_{lb1}$ and M_{lb2} , are compared with the exact values of M in 100 random trials, in which the semi-axis lengths and rotation angles of the ellipsoids are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. 136

5.6 With different aspect ratios of semi-axis lengths, errors of M_{ub} and M_{lb} are shown on a 2D grid plot, with the axes of b/a and c/a , where $b/a, c/a \in [1, 5]$ 137

5.7 Upper and lower bounds, F_{ub1}, F_{ub2} and F_{lb} , are compared with the exact values of F in 100 random trials, in which the semi-axis lengths and rotation angles of the ellipsoids are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. 139

LIST OF FIGURES

5.8 With different aspect ratios of semi-axis lengths, errors of F_{ub} and F_{lb} are shown on a 2D grid plot, with the axes of b/a and c/a , where $b/a, c/a \in [1, 5]$ 140

5.9 Comparisons of different bounds and exact values of Minkowski sums of two ellipses in two different cases. (a) Both ellipses are circles with radius 3 and 1.5, respectively. In this case, $V_{BM} = V_{lb} = V_{ub} = V_{exact} = \pi(3 + 1.5)^3 = 63.61$. (b) The ellipses have semi-axis lengths 1, 5 and 3, 6, and rotation angles 0 and $\pi/4$, respectively. $V_{BM} = 131.9 < V_{lb} = 165.2 < V_{exact} = 171.1 < V_{ub} = 179.1$ 142

5.10 The upper and lower bounds, V_{ub} and V_{lb} , compared with the exact value of the Minkowski-sum volumes of two ellipses, V_{exact} , based on 100 trials, in which the semi-axis lengths and angles of the ellipses are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. The relative errors of the upper and lower bounds, k_{ub} and k_{lb} , are calculated and shown in the figure Here, $k_{exact} = 0$ 143

5.11 Comparisons of different bounds and exact values of Minkowski sums of two ellipsoids in three different cases. (a) Both ellipsoids are spheres with radius 4 and 2.5, respectively. In this case, $V_{BM} = V_{lb1} = V_{lb2} = V_{exact} = V_{ub2} = V_{ub1} = \frac{4\pi}{3}(4 + 2.5)^3 = 1.15 \times 10^3$. (b) One ellipsoid is a sphere with radius 2, the other one is axis-symmetric with semi-axis lengths 2, 2, 6. $V_{BM} = 448.1 < V_{lb1} = V_{lb2} = V_{exact} = V_{ub2} = 563.3 < V_{ub1} = 573.7$. (c) Both ellipsoids are triaxial with semi-axis lengths as 7, 3, 3 and 2, 3, 4, and ZXZ Euler angles as $-\pi/4, -\pi/8, \pi/4$ and $\pi/3, \pi/4, -\pi/6$, respectively. $V_{BM} = 1354.4 < V_{lb1} = 1678.8 < V_{lb2} = 1724.3 < V_{exact} = 1725.9 < V_{ub2} = 1729.2 < V_{ub1} = 1741.2$ 146

5.12 The upper and lower bounds, $V_{ub1}, V_{ub2}, V_{lb1}$ and V_{lb2} , are compared with the exact value of the Minkowski-sum volumes, V_{exact} , based on 100 trials, in which the semi-axis lengths and angles of the ellipses are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. The relative errors of these bounds, $k_{ub1}, k_{ub2}, k_{lb1}$ and k_{lb2} , are calculated and shown in the figure Here, $k_{exact} = 0$ 147

5.13 The relative errors of different area estimates, V_{AM}, V_{GM} and V_{est} , for $E_1 \oplus E_2$ are calculated based on 100 trials, in which the semi-axis lengths and angles of the ellipses are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. 148

5.14 The relative errors of different volume estimates, V_{AM}, V_{GM} and V_{est} , for $E_1 \oplus E_2$ are calculated based on 100 trials, in which the semi-axis lengths and angles of the ellipsoids are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. 149

5.15 A planar example when one ellipse can move freely at all the orientations inside another without collision. 153

LIST OF FIGURES

5.16	A spatial example when one ellipsoid can move freely at all the orientations inside another without collision.	154
6.1	(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. . .	163
6.2	The detection scheme for the collision-free line segments on each sweep line using the logic in Eq. 6.1.	164
6.3	(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.	165
6.4	The collision-free space using the detection scheme for the example in Fig. 6.3.	165
6.5	The rabbit-shaped robot with 5 degrees of freedom — 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 $0 \leq \alpha_1 \leq 2\pi$, $0 \leq \alpha_2 \leq \pi/2$, and $-\pi/2 \leq \alpha_3 \leq 0$. . .	168
6.6	An example of generating the highway roadmap for the rabbit-shaped robot with 2 DOF (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).	168
6.7	(a) An example of the highway roadmap for the robot with 5 DOF. 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). (b) The steps of the motion of the robot overlaid onto the shortest path.	169
6.8	The 3D highway system for the example in Fig. 6.3.	169
6.9	The procedure of removing the redundant nodes on each single route. . . .	171
6.10	An example of a planar roadmap before and after removing the redundant nodes.	172
6.11	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 $-\pi/2 \leq \alpha_3 \leq 0$. . .	174

LIST OF FIGURES

6.12	A 3D example is shown with the narrow passage problem. The red and green rabbits represent the starting point and the goal, and each magenta rabbit represents a step along the feasible path connecting the starting point and the goal. The path is generated by the highway roadmap approach. . . .	177
6.13	The minimum distance along the paths. The black square box highlights where the minimum distance significantly decreases and the narrow passage problem occurs.	178
6.14	(a) The 5-link robot arm in the planar case with 7 DOF; (a) The 5-link robot arm in the spatial case with 18 DOF.	180
6.15	(a) The 2D maze environment; (b) The roadmap constructed for the 2D maze environment by using our safety function-based hybrid approach; (c) A feasible path found for the 2D maze environment.	181
6.16	An example of one “slice” of the safety function database with $\theta = 0$ for a single elliptical link in the 5-link planar articulated robot. The safety function database is 3D for planar cases with each slice representing a collision-free C-space with a fixed rotation. The black shaded regions represent the collision-free C-space while the white-shaded regions represent the C-space obstacles.	182
6.17	(a) A spatial path planning example is formed by a 5-link articulated robot roaming in an ellipsoidal environment clustered with ellipsoidal obstacles. The initial state (green) and the final state (red) of this 5-link articulated robot are defined and the robot has to pass a narrow passage to connect the initial state and the final state. (b) A roadmap of this path planning example is constructed by the PRM using 2000 nodes. In the roadmap, we can see that a narrow passage (with less nodes and connected edges) arises near the gap of the two small ellipsoidal obstacles.	183
6.18	Using the PRM combined with our safety function-based detection scheme, a feasible path is found. Some of the intermediate steps are illustrated. . . .	184
A.1	The GUI of group calculations. The GUI can provide four basic functions — subgroup test, normal subgroup test, group table generation and subgroup table generation.	196

Chapter 1

Introduction

1.1 Overview

Path planning originated in the field of robotics to plan collision-free paths for robots. As its name suggests, the goal of path planning is to plan paths. That is, given a description of an environment with known obstacles and a movable object, the goal is to compute a path of valid intermediate configurations that transforms the movable object from a given initial configuration (the start) into some desired final configuration (the goal). Motion planning arises not only in robotics (e.g., mobile robots [1] or robotic manipulators [2]), but also in diverse application domains such as CAD (e.g., maintainability studies [3, 4] and virtual prototyping [3]), computer animation [5, 6] and computational biology and chemistry (e.g., drug docking [7], protein folding [8] and protein crystallography [9–12]).

The primary focus of this dissertation is on geometric motion planning. The central

CHAPTER 1.

theme is by incorporating the useful geometrical models into the study of the motion space of the moving objects to obtain a better characterization of the collision-free configuration spaces and therefore to improve the efficiency in motion planning. Two different approaches are addressed in the contexts of biological crystallography and robot motion planning, respectively. Additionally, the quality of the sampling of the rotation and motion space is sometimes critical in these applications, especially for sampling based motion planning methods. In this dissertation, we also developed a geometric sampling approach to achieve almost-uniform sampling of the motion space and to further facilitate a broader class of motion planning problems. We classify the main content of this dissertation into three parts,

- One motion planning approach is proposed in the context of biological crystallography. The method of molecular replacement (MR) is frequently used to obtain phase information for a crystallographic unit cell packed with copies of a macromolecule of unknown conformation. Molecular replacement (MR) is a well-established computational method for phasing in macromolecular crystallography. This is important because an X-ray diffraction experiment on its own does not provide full structural information. The shape and symmetry of the unit cell is determined by the space group symmetry. It works quite well for single-domain proteins that can be treated as rigid-bodies. However, for multi-domain structures and complexes, computational requirements can become prohibitive due to the “curse of dimensionality”. We introduce a new approach to obtain the phase information by using geometric constraints

CHAPTER 1.

of protein crystals to generate packing models and therefore replace the computationally expensive MR searches in continuous configuration space with a search on a relatively small discrete set of candidate packing arrangements of a multi-rigid-body model.

- Another motion planning approach is based on the closed-form representation of the collision-free configuration spaces (C-spaces) of the robot. In this approach, the robot, obstacles, and environment are approximated as ellipsoids or finite unions of ellipsoids in the workspace. These objects can be quite general, including nonconvex bodies, and this approach represents an alternative to polyhedral representations of bodies. In the case of the robot, articulated degrees of freedom between unions of subsets of a larger set of ellipsoids can be included. The approach builds on the core idea that the space of collision-free motions of one moving ellipsoid relative to one fixed ellipsoid can be characterized completely and exactly in closed form. This means that for ellipsoidal bodies, there is no need to sample and discard poses suspected of being in collision. This approach of “knowing where to look” for path planning in C-spaces is especially useful for the narrow passage problem.
- We also propose an approach to achieve almost-uniform sampling on the rotation and motion spaces. It is a group-theoretic approach based on decomposing the rotation and Special Euclidean groups into identical Voronoi cells centered on the elements of their discrete subgroups. Within each cell, Cartesian grids in exponential coordinates

CHAPTER 1.

are used to achieve almost-uniform sampling at any level of resolution, without having to store large numbers of coordinates, and without requiring sophisticated data structures. We also analyze the shape of these cells, and discuss how this new method can be used in the context of conformational searches in the field of structural biology and robotics contexts.

Those three parts are self-contained but also interconnected. For example, the macromolecules or domains of macromolecules are often approximated by ellipsoidal representations [13]. The closed-form representation of the collision-free C-spaces based on the ellipsoidal models can be naturally applied to the macromolecular crystallography problem we addressed above. Furthermore, for both motion planning approaches, the discretization on the rotation space is required. The quality of the grid affects both the performance and reliability of planning algorithms. Having a uniform sampling will prevent search algorithms from oversampling or undersampling large portions of the C-spaces and therefore computational resources can be efficiently used.

1.2 Review of Motion Planning Methods

Motion planning is a broad problem with applications in many different contexts. The fundamental problem of all of its facets is to find valid motions for a robotic system to move from one configuration to another. This problem can be represented in the space of all possible configurations of the robot as the problem of finding a valid path between two

CHAPTER 1.

configurations. A valid path means that all parts of the path lie in the robot's collision-free C-space, i.e., the set of all robot configurations that does not violate any constraints posed by the motion planning problem at hand. In the rest of this dissertation, we use the term robot in a more general sense, to refer to any object that moves in either a 2D or 3D Euclidean space. First we provide a brief review on some of the most common motion planning approaches.

- Bug Algorithms

Bug path-planning algorithms, which are inspired by insects, assume that there is no global model of the world, i.e., obstacles are unknown, and only local information can be acquired through sensing. In these algorithms, the environment is a two-dimensional scene filled with unknown obstacles, with the robot represented as a point. Each obstacle is a simple closed curve of finite length and non-zero thickness. These algorithms are complete, i.e., they can find a solution if it exists, and report no when there is no solution. Bug 1 and Bug 2 [14] are probably the most straightforward path planning algorithms. By assuming essentially tactile (contact) sensing, the algorithms are to move toward the goal, unless an obstacle is encountered, in which case, move along the obstacle's boundary until motion toward the goal is once allowable again. Tangent Bug [15] serves as improvement to Bug 2 [14] by equipping the robot with a range sensor so that it can determine a shorter path to the goal.

- The Configuration Space

CHAPTER 1.

While bug algorithms do the path planning by assuming the robot as a point, when the robot has a geometric shape, the notion of the configuration space (or C-space) is needed. The configuration of a robot system is a complete specification of the position of every point of that system and the configuration space is the space of all possible configurations of the robot. The number of degrees of freedom (DOF) of a robot system is the dimension of the configuration space. The collision-free path can be planned with the knowledge of the collision-free configuration space or configuration space obstacles. The classic results for planning a safe path for a polyhedral robot moving among known polyhedral obstacles are shown in [16] and the robot is extended to the geometry of linked polyhedra (with more DOF) in [17]. The volume of the collision-free configuration space is characterized in [18]. The algorithm using configuration space is further implemented to plan collision-free motions for general manipulators, such as the Puma robot arm in [19]. In [20] the Fast Fourier Transform (FFT) was used to compute this convolution, especially for the workspaces with many complicated obstacles and/or the robot with a complicated shape. Some algorithms for fast convolutions on motion groups were studied in [21].

- Potential Functions

Explicitly representing the configuration space can be difficult. An alternative is to have an algorithm that incrementally explores free space while searching for a path. Bug algorithms can handle the free-configuration spaces without constructing them, but they are only limited in two-dimensional spaces. Navigation planners using po-

CHAPTER 1.

tential functions can be applied to a more general class of configuration spaces, including those that are multi-dimensional and non-Euclidean. The simplest potential function is the attractive/repulsive potential function [22]— the goal attracts the robot while the obstacles repel it. Or gradient descent algorithms can be used to move the robot opposite to the gradient. However, these algorithms are generically only guaranteed to converge to a local minimum, rather than a global minimum. The wave-front planner in [23] and [24] provides the simplest solution to the local minima problem, but can only be implemented in spaces that are represented as grids. The authors of [25] and [26] introduce the navigation function, a new potential that is a function of distance to the obstacles and has only one minimum to solve the local minima problem. They present a formula for navigation functions that guide a point-mass robot in a generalized sphere world and is applicable to configuration spaces whose forbidden regions can be modeled by generalized discs. A new potential field method is presented for non-spherical single-body robots by using the model that simulates steady-state heat transfer and minimizing the thermal resistance in a heat flow [27]. Instead of using a deterministic approach, the local minima can also be escaped by adding random motions [28]. For the robot in unknown and dynamic environments, the global dynamic window approach is developed in [29], which combines the dynamic window approach for reactive obstacle avoidance with the local minima-free navigation function to achieve high-speed motion planning.

- Mappings

CHAPTER 1.

If we knew that many paths were to be planned in the same environment, then it would make sense to construct a data structure (mapping) to plan subsequent paths more quickly. Mapping is the task of generating models of robot environments from sensor data. Mapping is important when the robot does not have *a priori* information about its environment and must rely on its sensor to gather information to incrementally construct its map. Topological representations encode environments with graph like structures, e. g. , nodes and edges. One particular topological map is the generalized Voronoi graph (GVG) [30], [31], [32], [33]. It is difficult to build an accurate topological map in a global coordinate frame, through exploration in an unknown unstructured environment. The authors of [34], [35], [36], [37], [37] and [38] add metric information (e.g., relative distance, angle) to the topological maps to help distinguish places that look the same. A very important class of topological maps is roadmaps [16], [39]. A roadmap is embedded in the free space and the nodes and edges of a roadmap also carry physical meaning. For example, a roadmap node can correspond to a specific location and an edge can be a path between neighboring locations. Geometric models use some basic geometric structures to represent the environment. Many researchers use line segments to represent parts of the environment [40]. More recent approaches use triangle meshes to represent three-dimensional structures of the environment [41], [42], [43]. Grid structures can also be used to describe the environment. In [44] the authors develop an approach based on the concept of frontiers (in grids), regions on the boundary between open

CHAPTER 1.

space and unexplored space. By moving to new frontiers, a mobile robot can extend its map into new territory until the entire environment has been explored. In [45] sonar range data was used to build a description of the robot's environment. Sonar readings are interpreted using probability profile to determine empty and occupied areas.

- Sampling-Based Algorithms

As the dimension of the configuration space grows, it is difficult to have an explicit representation of the configuration space. But this problem can be solved using sampling-based algorithms. Sampling-based methods use a variety of strategies for generating samples (collision-free configuration of the robot) and planning paths through connecting those samples. It is computationally efficient and easy to implement, but does not guarantee completeness. Instead, some of them offer probabilistic completeness, i.e., when a solution exists, the planner finds a solution as time goes to infinity, and when a solution does not exist, it may not be able to determine that a solution does not exist.

Probabilistic Roadmaps (PRM) were introduced in [46]. The basic PRM can be divided into two phases: the learning phase, when a road map is built, in which vertices correspond to collision-free configuration of the robot and two vertices are connected by an edge if a path between the two configuration is collision free; and the query phase, when user-defined query configuration are connected with the precom-

puted roadmap. Some applications of PRM in high-dimensional configuration spaces can be seen in [47] and [48]. PRM framework developed for planning collision-free geometric paths is extended to motion planning with both kinodynamic constraints and moving obstacles in [49]. Some variants of the basic PRM are developed such as Lazy PRM [50], visibility-based PRM [51], and PRM for star-shaped roadmaps [52]. Expansive-Spaces Tree (EST) is initially developed as an efficient single-query planner that covers the space between the initial configuration and the goal [49]. Another tree algorithm is called rapidly-Exploring Random Trees (RRT) [53–56]. The basis of RRT is to pull the tree toward random samples in the configuration space. It relies on nearest neighbors and a distance metric and adds Voronoi bias to tree growth. An extension of RRT, ERRT (execution extended RRT), introduces two extensions of the traditional RRT, the way point cache and adaptive cost penalty search, which improve replanning efficiency and the quality of generated paths [57].

One of the major goals of this dissertation is the parameterization of the collision-free space. This *a priori* knowledge of free space can be combined with the existing sample-based path planning methods (i.e., RRT and PRM) to facilitate the path search speed.

1.3 Contributions

The major contributions of this work are as follows,

- In Chap. 3, in the context of macromolecular crystallography, we propose an alter-

CHAPTER 1.

native to the traditional molecular replacement method, called phasing by packing, which is based on building packing models for macromolecules in crystal and generating a relatively small discrete set of candidate packing arrangements of a multi-rigid-body model, instead of computationally expensive MR searches in continuous configuration space. It is promising for multi-rigid-domain structures and numerical results demonstrate the potential of this method.

- In Chap. 4, we develop a sampling approach based on Voronoi cell decompositions in the rotation and motion groups. We make a connection between Voronoi cells in the special orthogonal group $SO(3)$ and the special Euclidean group $SE(2)$ centered on elements of discrete subgroups. We show that sampling within these Voronoi cells can be made almost uniform by exponentiating a Cartesian grid in a region of the corresponding Lie algebra, which is the pre-image of these cells under the exponential map. We demonstrate how the resulting cells, and the samples therein, can be used for searches and planning on certain Lie groups of interest in the contexts of robotics and computational biology.
- In Chap. 5, we developed an approach to parameterizing the exact boundaries of the Minkowski sum and difference of two ellipsoids. We also develop two other methods to quickly obtain the approximated Minkowski sum and difference based on the offset curve/surface approximations and special properties of the Gaussian distribution. In contrast to most existing methods, our approaches are completely

CHAPTER 1.

analytical and have closed forms and therefore naturally provide improved efficiency and better robustness in many applications. Based on Steiner's Formula, we also develop a method to provide the upper and lower bounds of the Minkowski-sum volumes, which in general do not have exact closed-form expressions.

- In Chap. 6, based on the closed-form representations of the Minkowski sum and difference of two ellipsoids, we develop a motion planning approach by describing the robot, obstacles, and environment as ellipsoids or finite unions of ellipsoids in the workspace. If the robot is a single-rigid-body, parametric representations of the collision-free region C-space are given and a “highway” roadmap system is constructed to connect the collision-free regions. Therefore collision checking can be eliminated not only for the vertices, but also for the edges. We can also use the knowledge of collision-free regions for single-rigid-bodies in path planning of articulated bodies. In this case, we construct a “safety function” to rapidly evaluate the collision of the articulated body by checking each single rigid link using the precomputed *a priori* knowledge. This “safety function” approach alleviates the traditional time-consuming collision checking for high-DOF articulated bodies. It can be integrated with any existing sampling-based path planning methods and improve the computation speed, particularly in the narrow passage problem.

Chapter 2

Mathematical Overview

In the chapter, we provide an overview of the major mathematical concepts that will be covered in the dissertation. These concepts form the common foundation for the novel approaches presented in this dissertation.

2.1 Basic Concepts in Group Theory

Groups:

In this thesis, we use group theory as a fundamental tool to study different crystallography symmetries as well as represent rigid-body rotations and motions. Here we first give a brief review of some basic concepts in group theory.

A group is a nonempty set G together with a binary operation \circ defined for pairs of elements of G which satisfy the following axioms:

CHAPTER 2.

1. Closure: If $g_1, g_2 \in G$, then $g_1 \circ g_2 \in G$.
2. Identity: There exists an identity element, e , such that $g \circ e = g$, for every $g \in G$.
3. Inverse: For each $g \in G$, there exists a (unique) inverse, $g^{-1} \in G$, such that $g \circ g^{-1} = e$.
4. Associativity: If $g_1, g_2, g_3 \in G$, then $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$.

The main kind of groups we used in this thesis is matrix Lie groups. A matrix Lie group (G, \circ) is a group whose elements are square matrices of a fixed dimension which is also a smooth manifold and the group operations $a(g_1, g_2) = g_1 \circ g_2$ and $b(g) = g^{-1}$ are smooth. The dimension of a Lie group is the dimension of the associated manifold G , rather than the dimension of the matrices.

A set \mathcal{G} is called the matrix Lie algebra that can be associated with G . Elements of \mathcal{G} can be obtained by taking the matrix logarithm of elements of G , when G is invertible. Given a matrix Lie group, elements sufficiently close to the identity are written as $g(t) = \exp(tX)$, for some $X \in \mathcal{G}$ and t near 0. For matrix Lie groups, the corresponding Lie algebra is usually denoted with lowercase letters [58]. Here are some examples of matrix Lie groups and matrix Lie algebras that are frequently used in this thesis.

Special orthogonal group $SO(n)$ in \mathbb{R}^n is defined as

$$SO(n) \doteq \{R \in \mathbb{R}^{n \times n} \mid RR^T = I, \det R = +1\}. \quad (2.1)$$

We will be primarily interested in $n = 2$ (planar rigid-body rotations) and $n = 3$ (spatial rigid-body rotations). The corresponding Lie algebra of $SO(n)$ is $n \times n$ skew-symmetric

CHAPTER 2.

matrices, i.e.,

$$so(n) \doteq \{\Omega \in \mathbb{R}^{n \times n} | \Omega^T = -\Omega\}. \quad (2.2)$$

In the instance of $SO(3)$,

$$\Omega = \hat{\omega} \doteq \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \quad (2.3)$$

with $\omega = [\omega_1, \omega_2, \omega_3]^T$.

The inverse operation of ‘ $\hat{\cdot}$ ’ used in (2.3) is ‘ \vee ’, i.e.,

$$(\hat{\omega})^\vee = \omega. \quad (2.4)$$

Special Euclidean group $SE(n)$, when $n = 2, 3$, is associated with rigid-body motions in planar and spatial cases. $SE(3)$, for example, is define as

$$SE(3) \doteq \{(R, \mathbf{t}) | R \in SO(3), \mathbf{t} \in \mathbb{R}^3\}, \quad (2.5)$$

with the group operation define as

$$g_1 \circ g_2 \doteq (R_1, \mathbf{t}_1) \circ (R_2, \mathbf{t}_2) = (R_1 R_2, R_1 \mathbf{t}_2 + \mathbf{t}_1). \quad (2.6)$$

Analogous to the definitio of $so(3)$, we defin the Lie algebra of $SE(3)$ as

$$se(3) \doteq \{(\hat{\omega}, \mathbf{v}) : \mathbf{v} \in \mathbb{R}^3, \hat{\omega} \in so(3)\}. \quad (2.7)$$

In homogeneous coordinates, we write an element $\hat{\xi} \in se(3)$

$$\hat{\xi} = \begin{pmatrix} \hat{\omega} & \mathbf{v} \\ \mathbf{0}^T & 0 \end{pmatrix} \in \mathbb{R}^{4 \times 4}, \quad (2.8)$$

CHAPTER 2.

where $\xi = [\xi_1, \dots, \xi_6]^T = [\omega^T, \mathbf{v}^T] \in \mathbb{R}^6$. An element of $se(3)$ is referred to as a twist in robotics.

Subgroups:

A subgroup is a subset of a group ($H \leq G$) which is itself a group that is closed under the group operation of G . The rotation group and special Euclidean group both contain discrete subgroups. In the case of the rotation group, these subgroups are the chiral point groups, and in the case of the special Euclidean group, the discrete subgroups are the chiral crystallographic space groups. A chiral symmetry group refers to one that preserves orientation, or equivalently, the right-handedness of coordinate systems. In Chap. 4, we use these discrete subgroups as “points” in Voronoi decomposition to divide the groups into identical units. For example, the discrete subgroups for $SE(2)$ are the five chiral wallpaper groups (see Fig. 2.1), and the discrete subgroups for $SO(3)$ are the rotational symmetry operations of the five Platonic solids (see Fig. 2.2).

Cosets, coset spaces and double cosets:

Given a subgroup $H \leq G$ and any fixed $g \in G$, the left coset gH is defined as

$$gH \doteq \{g \circ h | h \in H\}. \quad (2.9)$$

Similarly, the right coset Hg is defined as

$$Hg \doteq \{h \circ g | h \in H\}. \quad (2.10)$$

An important property of gH and Hg is that they have the same number of elements as H . Since the group is divided into disjoint cosets, each with the same number of elements,

CHAPTER 2.

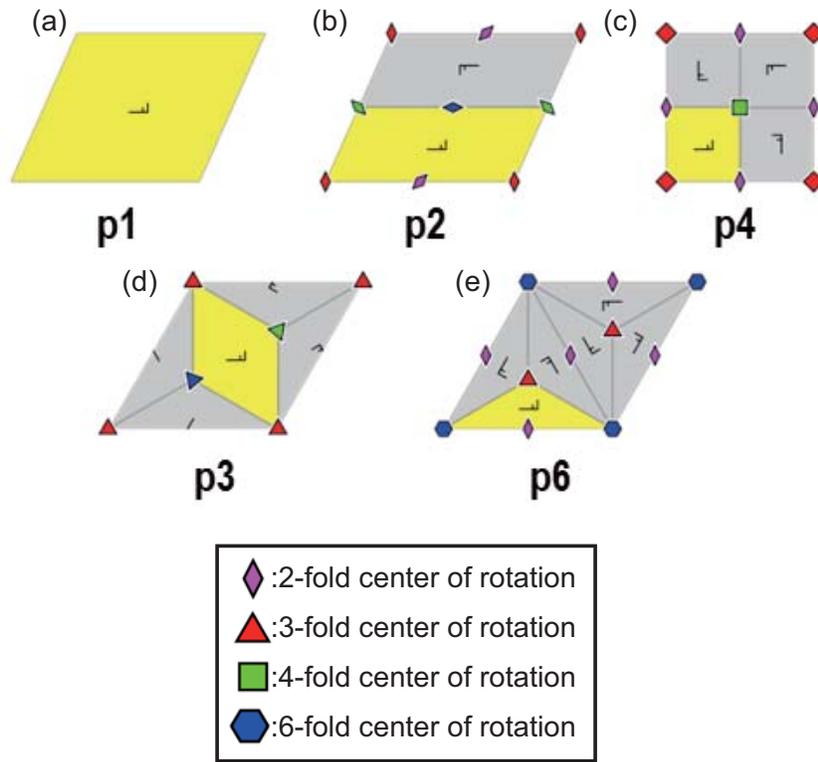


Figure 2.1: The five chiral wallpaper groups: (a) $p1$; (b) $p2$; (c) $p4$; (d) $p3$; (e) $p6$.

the number of cosets must divide without remainder the number of elements in the group, i.e., the group is divided into disjoint left (right) cosets. The set of all left (or right) cosets is called the left (or right) coset space and is denoted as G/H (or $H \backslash G$) with the property

$$|G/H| = |H \backslash G| = |G|/|H|. \quad (2.11)$$

This property is called Lagrange's group theorem. We note that when G is a finite group, $|\cdot|$ means the number of the group elements, and when G is a Lie group, it means volume.

Let $H < G$ and $K < G$. Then for any $g \in G$, the double coset of H and K is defined as

$$HgK \doteq \{h \circ g \circ k | h \in H, k \in K\}. \quad (2.12)$$

CHAPTER 2.

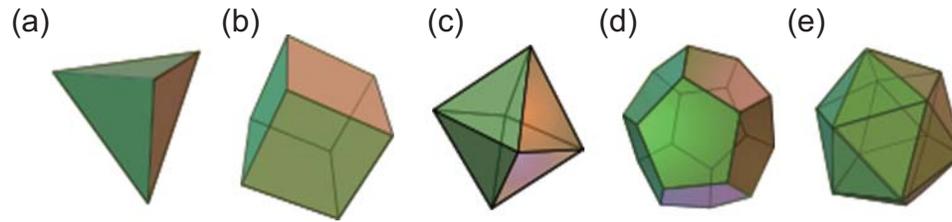


Figure 2.2: The five Platonic solids: (a) tetrahedron; (b) cube; (c) octahedron; (d) icosahedron (e) dodecahedron.

Normal subgroups and quotient groups:

A normal subgroup, N , is a normal subgroup of a group, G , denoted as $N \triangleleft G$, if it is invariant under conjugation, i.e.

$$gng^{-1} \in N, \quad \forall n \in N, \forall g \in G. \quad (2.13)$$

For a group G and a normal subgroup N of G , the quotient group of N in G , written G/N , is the set of cosets of N in G . When N is normal in G , left and right cosets are the same. Quotient groups are also called factor groups. The elements of G/N are written Na and form a group under the normal operation on the group N on the coefficient a . Thus,

$$(Na)(Nb) = Nab. \quad (2.14)$$

Since all elements of G will appear in exactly one coset of the normal subgroup N , it follows that

$$|G| = |G/N||N|, \quad (2.15)$$

where $|G|$ denotes the order of a group. This is also a consequence of Lagrange's group theorem with $H = N$.

Fundamental domains:

CHAPTER 2.

Associated with any (double-)coset, it is possible to define a set of distinguished (double-)coset representatives, exactly one per (double-)coset. Such a set defines a fundamental domain in G that has the same dimension as G , but lesser volume. Under the left action by H , the fundamental domain $F_{H \setminus G}$ is translated and the closure of the union of all translates covers G without measurable gaps or overlaps. Similarly, right action by K on the fundamental domain $F_{G/K}$ and the double-sided-action of $H \times K$ on $F_{H \setminus G/K}$ produces translates the closure of which cover G .

One way to construct fundamental domains is as Voronoi cells within G . Since G is a Riemannian manifold, a distance function $d : G \times G \rightarrow \mathbb{R}_{\geq 0}$ exists, and we can define

$$F_{H \setminus G} := \{g \in G \mid d(e, g) < d(e, h \circ g), \forall h \neq e \in H\}, \quad (2.16)$$

$$F_{G/K} := \{g \in G \mid d(e, g) < d(e, g \circ k), \forall k \neq e \in K\},$$

and when $H \cap K = \{e\}$, G is partitioned into disjoint fundamental domains for double coset spaces $F_{H \setminus G/K}$ where

$$F_{H \setminus G/K} \doteq \{g \in G \mid d(e, g) < d(e, h \circ g \circ k), \forall (h, k) \neq (e, e) \in H \times K\}. \quad (2.17)$$

Space groups:

In crystallography, there are 230 space groups, 65 of them chiral ones, and only these occur in protein crystallography. A wallpaper group is a plane crystallographic group. There are 17 possible distinct groups. 5 of them are chiral groups. Of the 65 chiral space

CHAPTER 2.

groups, 24 can be written as semi-direct products, and the fundamental domain for $\Gamma \backslash G$ can be written as

$$F_{\Gamma \backslash G} \cong F_{\mathbb{Z}^3 \backslash \mathbb{R}^3} \times F_{\mathbb{P} \backslash SO(3)}, \quad (2.18)$$

where $G = \mathbb{R}^3 \rtimes SO(3)$ and $\Gamma = \mathbb{Z}^3 \rtimes \mathbb{P}$ and $\mathbb{P} < SO(3)$ is the crystallographic point group and $\mathbb{Z}^3 < \mathbb{R}^3$ is the lattice translation group. These are called symmorphic, and the other 41 are called nonsymmorphic. In the planar case, all five chiral wallpaper groups are symmorphic.

In App. A, a graphical user interface (GUI) is developed for space group calculations. Specifically, given the coset representatives of a space group, all subgroups can be found and whether each of them is normal subgroup or not can be tested. It provides a fundamental calculation tool for some more advanced tests, such as normal Bieberbach and symmorphic subgroup test.

2.2 Rigid-body Rotation and Motion Spaces

A rigid-body motion of an object is a motion which preserves distance between points and handedness of vectors. The concept of the rigid-body motion is widely used throughout this thesis. From packing biomolecular models (Chap. 3) in crystals to robot motion planning (Chap. 5), we treat all objects as rigid-bodies. In Chap. 4, we also study the problem of uniform sampling of rigid-body rotation and motion spaces. Here, we review some of their basic concepts related to our applications.

CHAPTER 2.

Rigid-body rotation is a special case of a rigid-body transformation, which is defined as motions that preserve the distance between points in an object before and after the motion and leave one point fixed under the motion. As mentioned previously in Sec. 2.1, the spatial rigid-body rotation group is denoted as $SO(3)$. Several methods can be used to parameterize rotations including axis-angle description, Euler angles, quaternions, Cayley/Rodrigues parameters, Cartesian Coordinates in \mathbb{R}^4 , and parameterization as a solid ball in \mathbb{R}^3 [58]. Euler angles and solid ball (or exponential) parameterization are mainly used in this thesis.

Euler angles are by far the most widely known parameterization method of rotation. They are arranged by three successive rotations about independent axes. One of the most common choices and also the one used in this thesis is ZXZ Euler angles, denoted as

$$R_{ZXZ}(\alpha, \beta, \gamma) = R_z(\alpha)R_x(\beta)R_z(\alpha). \quad (2.19)$$

The corresponding matrix can be written explicitly as

$$R_{ZXZ} = \begin{pmatrix} \cos \gamma \cos \alpha - \sin \gamma \sin \alpha \cos \beta & -\sin \gamma \cos \alpha - \cos \gamma \sin \alpha \cos \beta & \sin \beta \sin \alpha \\ \cos \gamma \sin \alpha + \sin \gamma \cos \alpha \cos \beta & -\sin \gamma \sin \alpha + \cos \gamma \cos \alpha \cos \beta & -\sin \beta \cos \alpha \\ \sin \beta \sin \gamma & \sin \beta \cos \gamma & \cos \beta \end{pmatrix}. \quad (2.20)$$

The ranges of these angles are $0 \leq \alpha \leq 2\pi$, $0 \leq \beta \leq \pi$, and $0 \leq \gamma \leq 2\pi$.

Let a time-varying rotation matrix be parameterized as

$$R(t) = A(q_1(t), q_2(t), q_3(t)) = A(\mathbf{q}(t)), \quad (2.21)$$

CHAPTER 2.

the left and right Jacobians can be define as

$$\begin{aligned} J_L(A(\mathbf{q})) &= \left[\left(\frac{\partial A}{\partial q_1} A^T \right)^\vee, \left(\frac{\partial A}{\partial q_2} A^T \right)^\vee, \left(\frac{\partial A}{\partial q_3} A^T \right)^\vee \right], \\ J_R(A(\mathbf{q})) &= \left[\left(A^T \frac{\partial A}{\partial q_1} \right)^\vee, \left(A^T \frac{\partial A}{\partial q_2} \right)^\vee, \left(A^T \frac{\partial A}{\partial q_3} \right)^\vee \right]. \end{aligned} \quad (2.22)$$

The Jacobians for the ZXZ Euler angles are [59]

$$\begin{aligned} J_L(\alpha, \beta) &= \begin{pmatrix} 0 & \cos \alpha & \sin \alpha \sin \beta \\ 0 & \sin \alpha & -\cos \alpha \sin \beta \\ 1 & 0 & \cos \beta \end{pmatrix}, \\ J_R(\beta, \gamma) &= \begin{pmatrix} \sin \beta \sin \gamma & \cos \gamma & 0 \\ \sin \beta \cos \gamma & -\sin \gamma & 0 \\ \cos \beta & 0 & 1 \end{pmatrix}. \end{aligned} \quad (2.23)$$

Another way to represent rotations is as a solid ball in \mathbb{R}^3 with radius π in exponential coordinates. In this description, the radius of each concentric spherical shell within the ball represents a rotation angle. With a unit vector \mathbf{n} indicating the rotation axis and an angle θ describing the amount of rotation around \mathbf{n} , any rotation matrix $R \in SO(3)$ can be represented as

$$R = \exp(\hat{\mathbf{n}}\theta) = \mathbb{I} + \hat{\mathbf{n}} \sin \theta + \hat{\mathbf{n}}^2 (1 - \cos \theta), \quad (2.24)$$

where $\hat{\mathbf{n}}$ is the skew-symmetric matrix corresponding to \mathbf{n} (see (2.3)). This can be viewed as spherical coordinates in the Lie algebra $so(3) \cong \mathbb{R}^3$ in which θ takes the place of the radius. The corresponding Cartesian coordinates are $\mathbf{x} = \theta \mathbf{n}$ where $\theta = \|\mathbf{x}\|$ and $\mathbf{n} = \mathbf{x}/\|\mathbf{x}\|$. In the component

CHAPTER 2.

form,

$$\mathbf{x}(\theta, \alpha', \beta') = \theta \begin{pmatrix} \sin \alpha' \cos \beta' \\ \sin \alpha' \sin \beta' \\ \cos \alpha' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} n_1 \theta \\ n_2 \theta \\ n_3 \theta \end{pmatrix}, \quad (2.25)$$

where $0 \leq \theta \leq \pi$, $0 \leq \alpha' \leq \pi$, $0 \leq \beta' \leq 2\pi$ and $\mathbf{x} = [x, y, z]^T$ are the Cartesian coordinates for the points inside the ball. We can obtain the same rotation matrix in (2.24) by exponentiating the skew-symmetric matrix corresponding to $\mathbf{x}(\theta, \alpha', \beta')$. For this reason, it is also often called exponential parameterization. Here we use α' and β' to distinguish from the first two Euler angles α and β mentioned earlier.

The Jacobians for $SO(3)$ in exponential parameterization are

$$R(\mathbf{x}) = \exp(\hat{\mathbf{x}}) \quad (2.26)$$

are known as [59, 60]

$$\begin{aligned} J_L(\mathbf{x}) &= \mathbb{I} + \frac{1 - \cos \|\mathbf{x}\|}{\|\mathbf{x}\|^2} X + \frac{\|\mathbf{x}\| - \sin \|\mathbf{x}\|}{\|\mathbf{x}\|^3} X^2, \\ J_R(\mathbf{x}) &= \mathbb{I} - \frac{1 - \cos \|\mathbf{x}\|}{\|\mathbf{x}\|^2} X + \frac{\|\mathbf{x}\| - \sin \|\mathbf{x}\|}{\|\mathbf{x}\|^3} X^2, \end{aligned} \quad (2.27)$$

where \mathbf{x} is the Cartesian coordinates in exponential coordinates for $SO(3)$. $J_L(\mathbf{x})$ and $J_R(\mathbf{x})$ have properties that

$$J_L(\mathbf{x}) = J_R^T(\mathbf{x}) \quad \text{and} \quad J_L(\mathbf{x}) = R(\mathbf{x})J_R(\mathbf{x}). \quad (2.28)$$

For rigid-body motions, homogeneous representation is widely used. The transformation of points and vectors by rigid transformations has a simple representation in terms of matrices and vectors in \mathbb{R}^4 . For the spatial rigid-body motion group $SE(3)$, $g = (\mathbf{t}, R) \in SE(3)$ is defined as a

CHAPTER 2.

4 × 4 matrix

$$g = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad (2.29)$$

where $R \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$.

Also, it is important to define ‘intrinsic’ measures of distance of rigid-body rotations and motions. By using the matrix logarithm function, it becomes possible to define

$$d_{SO(3)}(R_1, R_2) = \|\log(R_1^T R_2)\| \quad (2.30)$$

when $R_1^T R_2$ is not a rotation by π , and otherwise $d_{SO(3)}(R_1, R_2) = \pi$ and similarly

$$d_{SE(3)}(g_1, g_2) = \|\log(g_1^{-1} \circ g_2)\|_W \quad (2.31)$$

where $W = W^T$ as a 3 × 3 positive definite weighting matrix.

2.3 Crystallographic Symmetry

In crystallography, symmetry is used to characterize crystals, identify repeating parts of molecules, and simplify both data collection and calculations. A macromolecular crystal is composed of regularly-repeating blocks called unit cells that have a discrete symmetry group, Γ , called the crystallographic space group. This symmetry group divides \mathbb{R}^3 into unit cells, $U \cong \Gamma \backslash \mathbb{R}^3$ and also describes how copies of the crystals are located within the unit cell. The whole group Γ can be generated by translating unit cells and moving within the unit cell using generators $\{\gamma_1, \dots, \gamma_m\}$. These form a subgroup of Γ , which is in turn a subgroup of the group of rigid-body motions, $SE(3)$, denoted here as G .

CHAPTER 2.

The crystallographic space groups have been studied in great detail in the crystallography literature. For example, summaries can be found in [61, 62] as well as in various online resources. Treatments of space group symmetry from the perspective of pure mathematicians can be found in [63].

Of the 230 space groups, only 65 are possible for biological macromolecular crystals (i.e., the chiral/proper ones). The reason for this is that biological macromolecules such as proteins and nucleic acids are composed of constituent parts that have handedness and directionality. This is discussed in greater detail in [64–66]. Of these 65, some occur much more frequently than others. And these are typically nonsymmorphic space groups (i.e., those that possess screw symmetry operations, and these symmetry groups cannot be described as a simple semi-direct product). For example, more than a third of all proteins crystallized to date have $P2_12_12_1$ symmetry, and the three most commonly occurring symmetry groups represent approximately half of all macromolecular crystals [66, 67].

A proper crystallographic space group is a discrete (but infinite) subset of the group of rigid-body motions. This relationship is denoted as $\Gamma < G$, both of which have the operation \circ . Every crystallographic space group has a lattice translation group as a normal subgroup, $T \triangleleft \Gamma$, where $T \cong P_1$. It follows that the coset space $T \backslash \Gamma = \Gamma/T$ is a group with elements that are cosets of the form $T\gamma_i \in T \backslash \Gamma$. Here $i = 1, \dots, |P|$ and $P = T \backslash \Gamma$ is the point group of the crystal. Depending on the symmetry type, this point group can be constructed to contain purely rotational elements (in which case it is called *symmorphic*), or otherwise it must contain a combination of rotations and discrete screw motions (in which case it is called *nonsymmorphic*). In either case, $T \backslash \Gamma$ is isomorphic to the group that can be constructed from representatives $\{\gamma_1, \dots, \gamma_{|P|}\}$. In the main part of this paper we

CHAPTER 2.

use the notation $m = |P|$.

Here, we give an example of the most common symmetry $P2_12_12_1$ in crystallized proteins. For $P2_12_12_1$, the $m = 4$ representative symmetry elements are $\{e, \gamma_1, \gamma_2, \gamma_3\}$ where their actions on points in \mathbb{R}^3 are $e \cdot (x, y, z) \doteq (x, y, z)$; $\gamma_1 \cdot (x, y, z) \doteq (-x + 1/2, -y, z + 1/2)$; $\gamma_2 \cdot (x, y, z) \doteq (-x, y + 1/2, -z + 1/2)$; and $\gamma_3 \cdot (x, y, z) \doteq (x + 1/2, -y + 1/2, -z)$. Computing $(\gamma_i \circ \gamma_j) \cdot (x, y, z) = \gamma_i \cdot (\gamma_j \cdot (x, y, z))$ and then removing translations corresponding to positions that are outside of the unit cell define the operation \square that turns the set of representatives of $T \setminus \Gamma$ into a group. The following group table summarizes the group $(T \setminus \Gamma, \square) = (P1 \setminus P2_12_12_1, \square)$:

\square	e	γ_1	γ_2	γ_3
e	e	γ_1	γ_2	γ_3
γ_1	γ_1	e	γ_3	γ_2
γ_2	γ_2	γ_3	e	γ_1
γ_3	γ_3	γ_2	γ_1	e

Note that this is not the group (Γ, \circ) , which is infinite

In Fig. 2.3, an articulated 3-body molecular model that looks like a 'rabbit' is packed a unit cell in $P2_12_12_1$ symmetry with the dimension $L \times L \times L$. Let the corner of the unit cell (the box) be located at $(0, 0, 0)$, then the coordinates of the center of the face of the first rabbit (the purple one) are $(L/4, L/4, L/4)$ where $L = 80$ represents the dimension of the unit cell. The transformations that produce replicas are then (x, y, z) ; $(L/2 - x, L - y, L/2 + z)$; $(L/2 + x, L/2 - y, L - z)$; $L - x, L/2 + y, L/2 - z$. Here we have modified slightly the standard choice of coset representatives so that all the four rabbits fit in this unit cell. In all subfigure the orientation of each 'ear' relative to the 'face' is kept constant, and in the figure the principal axes of the face are aligned with the axes of the unit cell. If each rabbit is rigidly moved while maintaining $P2_12_12_1$ symmetry, a new configuratio

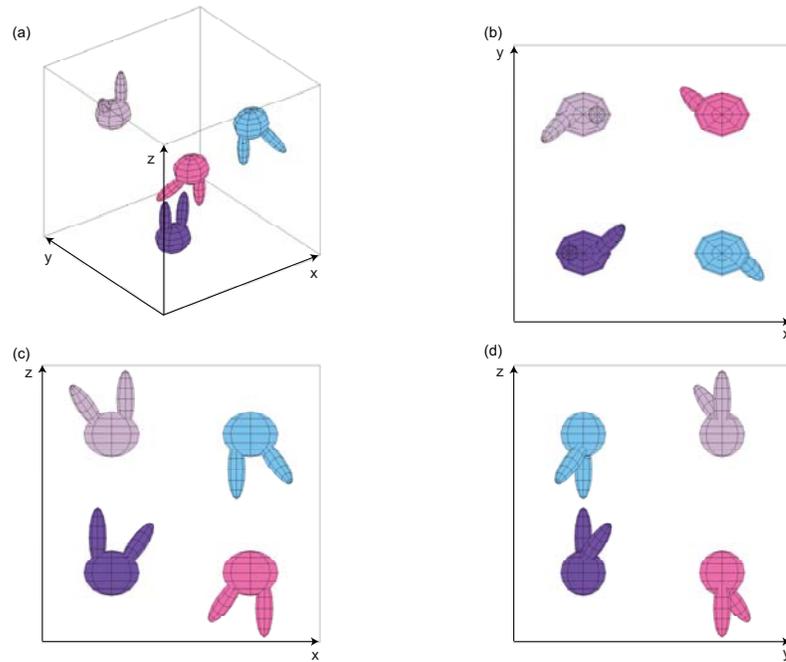


Figure 2.3: Objects arranged with $P2_12_12_1$ symmetry. (a) 3D view; (b)-(d) projections.

such as that in Fig. 2.4 will result. Here the translation vector of the purple copy is $[0, 2 - 2]^T$ and a rotation in ZYZ Euler angles of $\alpha = \pi/2$, $\beta = \pi/2$ and $\gamma = \pi/12$ is shown.

2.4 Voronoi Diagrams

Voronoi diagrams, one of the most fundamental data structures in computational geometry, have the importance in a wide variety of applications including facility planning, robot motion planning, computer graphics, epidemiology, geophysics, meteorology, etc [68–70]. Voronoi diagrams have received lots of attention because they have interesting mathematical properties which are related to many well-known geometrical structures [68, 70]. In recent years, they have proved to be a pow-

CHAPTER 2.

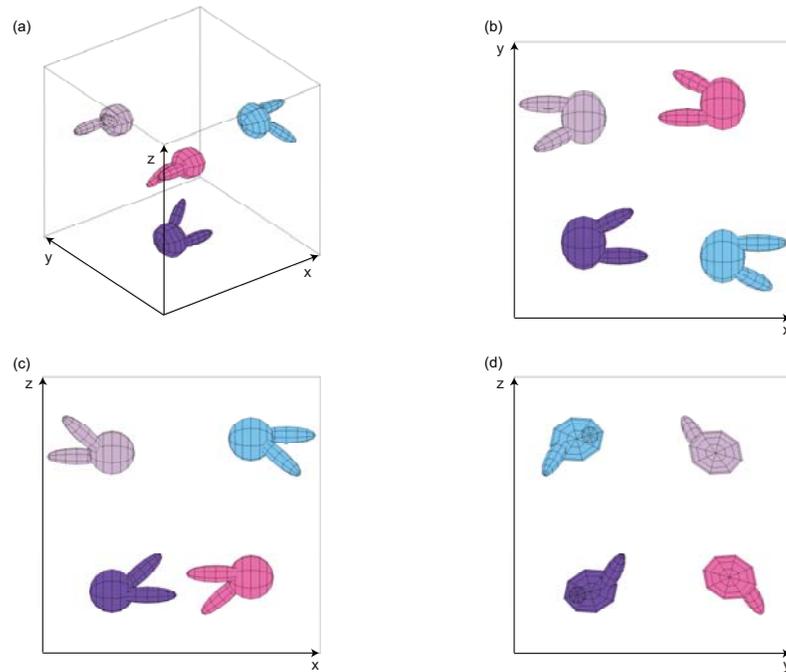


Figure 2.4: Coordinated movement of these objects that maintains $P2_1 2_1 2_1$ symmetry. (a) 3D view; (b)-(d) projections.

erful tool in solving seemingly unrelated computational problems. In Chap. 4, Voronoi diagrams, collaborated with coset- and double- coset decompositions, partition the rotation and motion spaces into identical units and facilitate almost-uniform sampling of these spaces. In Chap. 5 a “highway” roadmap system is constructed to connect different collision-free sites in the robot motion planning and the “highway” roadmap is essentially medial axes of the contours on some Voronoi diagrams. The formal mathematical definitio and some properties of Voronoi diagrams are reviewed as follows.

Let X be a space (a nonempty set) endowed with a distance function d . Let K be a set of indices and let $(P_k)_{k \in K}$ be a tuple (ordered collection) of nonempty subsets (the sites) in the space X . The Voronoi cell, or Voronoi region, S_k , associated with the site P_k is the set of all points in X whose

CHAPTER 2.

distance to P_k is not greater than their distance to the other sites P_j , where j is any index different from k . In other words, if $d(x, A) = \inf\{d(x, a) \mid a \in A\}$ denotes the distance between the point x and the subset A , then

$$S_k = \{x \in X \mid d(x, P_k) \leq d(x, P_j) \text{ for all } j \neq k\} \quad (2.32)$$

The Voronoi diagram is simply the tuple of cells $(S_k)_{k \in K}$ [70].

Some of the important properties of Voronoi diagrams includes [70, 71]

- The dual graph for a Voronoi diagram (in the case of a Euclidean space with point sites) corresponds to the Delaunay triangulation for the same set of points.
- The closest pair of points corresponds to two adjacent cells in the Voronoi diagram.
- Assume the setting is the Euclidean plane and a group of different points are given. Then two points are adjacent on the convex hull if and only if their Voronoi cells share an infinitel long side.
- If the space is a normed space and the distance to each site is attained (e.g., when a site is a compact set or a closed ball), then each Voronoi cell can be represented as a union of line segments emanating from the sites.

Chapter 3

A Phasing Approach Using Packing

Models for Multi-Domain Structures in Crystals

In the context of X-ray crystallography, the molecular replacement (MR) method is frequently used to obtain phase information for a crystallographic unit cell packed with a macromolecule of unknown conformation. This is important because an X-ray diffraction experiment on its own does not provide full structural information. The shape and symmetry of the unit cell is determined by the space group symmetry. The goal of MR searches is to place a homologous/similar molecule in the unit cell so as to maximize the correlation with X-ray diffraction data, and then to use the model to add the unknown phase information to the experimental data. MR software packages typically perform rotation and

CHAPTER 3.

translation searches separately. This works quite well for single-domain proteins that can be treated as rigid-bodies. However, for multi-domain structures and complexes, computational requirements can become prohibitive and the desired peaks can become hidden in a noisy landscape.

The main contribution of our approach is that computationally expensive MR searches in continuous configuration space are replaced by a search on a relatively small discrete set of candidate packing arrangements of a multi-rigid-body model. First, candidate arrangements are generated by collision detection on a coarse grid in the configuration space. For planar cases, we also design a Gaussian-based potential function. The candidate arrangements are generated by minimizing this potential function that forces the model conformations to separate from each other and not overlap within the unit cell. This set of feasible arrangements is small because collision-free packing requirement together with unit cell symmetry and geometry impose strong constraints. After computing Patterson correlations of the collision-free arrangements, an even shorter list can be obtained using the 10 candidates with highest correlations. In numerical trials, we found that a candidate from the feasible set is usually similar to the arrangement of the target structure within the unit cell. To further improve the accuracy, a rapidly-exploring random tree (RRT) can be applied in the neighborhood of this packing arrangement. Our approach is demonstrated with multi-domain models *in silico* for both planar and spatial cases, with ellipses and ellipsoids representing both the domains of the model and target structures. The numerical results illustrate the potential of this method in the context of $p1$ wallpaper group symmetry (for

planar cases) and $P1$ and $P2_12_12_1$ space-group symmetries (for spatial cases).

3.1 Introduction

The field of structural biology is concerned with characterizing the shape, composition, flexibility and motion of biological macromolecules as well as the complexes that they form. An ultimate goal of this field is to link these properties with macromolecular structures, in the hope of better understanding biological phenomena and designing new drugs.

Here we review some of the issues involved in translating experimental data into 3D structures in the context of protein crystallography. Macromolecular X-ray crystallography (MX) is the most widely used method for solving protein structures and associated complexes. In the X-ray crystallography experiment, only the diffraction pattern (the magnitude of the Fourier transform of the density of the contents of a crystallographic unit cell) can be collected and the phase information is lost (see the schematic diagram in Fig. 3.1). Therefore, it is not possible to directly reconstruct the protein's electron density map by the inverse Fourier transform of the diffraction pattern. The molecular replacement (MR) method is frequently used to obtain phase information for a unit cell packed with an unknown protein. The goal of MR searches is to place a homologous/similar molecule in the unit cell and search in the whole configuration space so as to maximize the correlation with X-ray diffraction data. See [12] and references therein for a complete description of the

CHAPTER 3.

MR method.

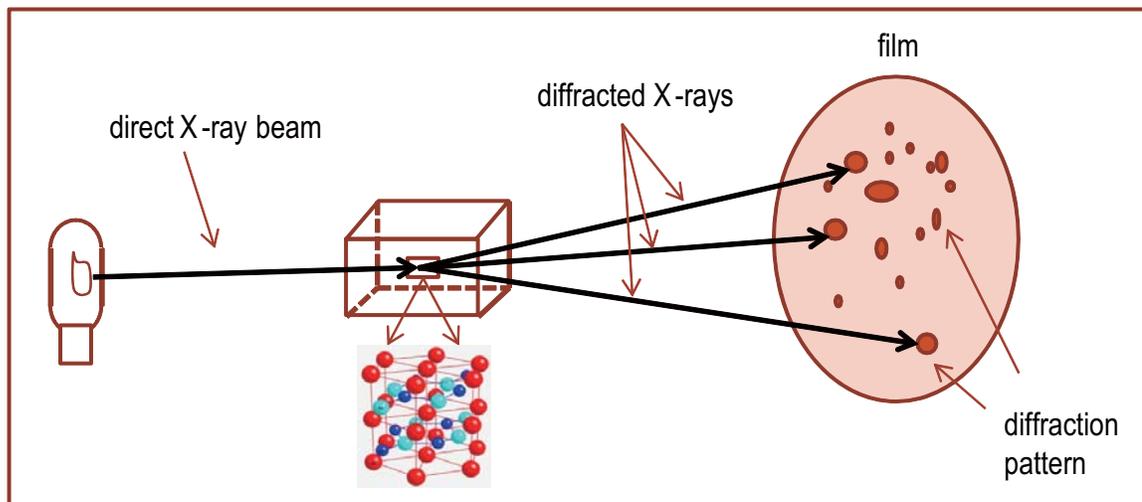


Figure 3.1: The schematic diagram of a macromolecular X-ray crystallography experiment. Only the diffraction data can be obtained (on the film) and the phase information of the molecule is killed.

The MR method works well for simple proteins that can be described as single-rigid-bodies (called domains). Much of this success relies on the information on the shape of $\sim 100,000$ previously solved structures in the Protein Data Bank (PDB), in which many of proteins are single-domain structures. The known single-domain structures are used to augment new MX experimental information to gain a complete picture of other proteins with similar structures.

However, a challenge to MX arises in interpreting X-ray diffraction patterns for crystals composed of multi-domain systems. Even when a multi-domain structure has been solved previously, its overall shape may vary widely from a new version of the structure with, for example, a bound drug. In this case, a widely used computational method called the

CHAPTER 3.

molecular replacement (MR) method, which has been highly successful for single-domain proteins, becomes combinatorially intractable due to the large number of degrees of freedom in multi-domain systems. We present a new method for phasing based on geometric packing that can serve as an alternative to MR. Decades ago, the concept of building models of crystallographic unit cells to phase crystallographic data was explored in the context of small molecules [72–74]. But to our knowledge, this approach has not been pursued and is virtually unknown in the context of multi-domain macromolecular crystallography, and “phasing by packing” therefore represents a very different way of approaching the problem than MR.

The remainder of this chapter is structured as follows. Sec. 3.2 provides an overview of mathematical aspects of macromolecular X-ray crystallography. Sec. 3.3 reviews the traditional MR method in finding the phase information for single-domain structures and also formulates the difficulties of the MR method for the multi-domain structures. Sec. 3.4 presents our new approach, called “phasing by packing”. This approach uses geometric constraints of crystal structures and packing models in crystal unit cells to find the phase information. It is especially promising for multi-rigid-domain structures because by using the packing models, we can replace the computationally expensive MR searches in high-dimensional continuous configuration space with a search on a relatively small discrete set of candidate packing arrangements. Two different methods are introduced to find the set of candidate packing arrangements — by collision detection (Sec. 3.4.1) and by minimizing a Gaussian-based potential function (Sec. 3.4.2). In Sec. 3.5, numerical results with multi-

domain toy models demonstrate the potential of this approach in the context of both the planar wallpaper symmetry $p1$, and the space group symmetries, $P1$ and $P2_12_12_1$. Finally, Sec. 3.6 is the conclusion.

3.2 Essentials of Macromolecular X-Ray Crystallography (MX)

A biological macromolecule is a large collection of atomic nuclei that are stabilized through a combination of covalent bonds, hydrogen bonds, and hydrophobicity. A traditional goal in structural biology is to obtain the Cartesian coordinates of all atoms in a rigid single-domain protein.

Let $\mathbf{x}_i = (x_i, y_i, z_i)$ denote the Cartesian coordinates of the i^{th} of n atoms in a single-domain protein structure, and let $\rho_i(\mathbf{x})$ be the electron density of that atom in a reference frame centered on it. Due to thermal motions, the electron density of each of these atomic nuclei can be treated as a Gaussian distribution. The density of the whole structure is then of the form

$$f(\mathbf{x}) = \sum_{i=1}^n \rho_i(\mathbf{x} - \mathbf{x}_i). \quad (3.1)$$

The coordinates $\{\mathbf{x}_i\}$ are typically given either in a reference frame attached to a crystallographic unit cell, or to the center of mass of the protein.

MX does not provide $f(\mathbf{x})$ directly. Rather, it provides partial information about $f(\mathbf{x})$.

CHAPTER 3.

The goal is then to computationally obtain $f(\mathbf{x})$ and fit an atomic model to it, thereby extracting the coordinates $\{\mathbf{x}_i\}$. A macromolecular crystal is composed of *unit cells* that have a discrete symmetry group, Γ , called the crystallographic space group. This symmetry group divides \mathbb{R}^3 into unit cells, $U \cong \Gamma \backslash \mathbb{R}^3$ and also describes how copies of the density $f(\mathbf{x})$ are located within the unit cell. The whole group Γ can be generated by translating unit cells and moving within the unit cell using generators $\{\gamma_1, \dots, \gamma_m\}$. These form a subgroup of Γ , which is in turn a subgroup of the group of rigid-body motions, $SE(3)$, which will be denoted here as G . The group $\Gamma = P2_12_12_1$ is of particular importance because roughly one third of all biological macromolecules that have been crystallized to date have this symmetry.

The result of an MX experiment is a diffraction pattern. This is the magnitude of the Fourier transform of the full contents of the crystallographic unit cell. Mathematically, this is written for a single-domain protein as

$$\hat{P}(g; \mathbf{k}) = \left| \mathcal{F} \left(\sum_{j=0}^{m-1} f((\gamma_j \circ g)^{-1} \cdot \mathbf{x}) \right) \right|, \quad (3.2)$$

where $|\cdot|$ denotes the modulus of a complex number, $c = a + ib = |c|e^{i\phi}$. Our reason for using the notation $\hat{P}(g; \mathbf{k})$ will be explained shortly. Here $g \in G$ is the unknown pose of the protein that is sought, and \circ is the group operation for both G and Γ . In particular, it is well-known in robotics that each rigid-body motion consists of a rotation-translation pair $g = (R, \mathbf{t})$, and the composition of any two rigid-body motions g_1 and g_2 define the operation \circ :

$$g_1 \circ g_2 = (R_1, \mathbf{t}_1) \circ (R_2, \mathbf{t}_2) = (R_1 R_2, R_1 \mathbf{t}_2 + \mathbf{t}_1). \quad (3.3)$$

CHAPTER 3.

Given that $g = (R, \mathbf{t}) \in G$ is a rotation-translation pair, its action on \mathbb{R}^3 is defined by

$$g \cdot \mathbf{x} = R\mathbf{x} + \mathbf{t}. \quad (3.4)$$

Then the density of a collection of single-domain proteins in the unit cell for $j = 0, \dots, m-1$ will be $\sum_{i=0}^{m-1} f((\gamma_i \circ g)^{-1} \cdot \mathbf{x})$.

The difficulty in extracting $f(\mathbf{x})$ from the MX data is that this measurement folds in both information about $f(\mathbf{x})$ and the symmetry group Γ , and loses the phase information, $\phi(\mathbf{k})$, without which $f(\mathbf{x})$ cannot be recovered by an inverse Fourier transform. Moreover, there is an unknown $g \in G$ that describes how each symmetry-related copy of $f(\mathbf{x})$ sits in the unit cell. Single-domain MR is mostly about finding the unknown g , and most commonly this is done by dividing the search into rotational and translational parts.

The crystallographic space groups have been studied in great detail in the crystallography literature. For example, summaries can be found in [61, 62] as well as in various online resources. Treatments of space group symmetry from a mathematician's perspective can be found in [63].

The number of proteins in a unit cell, the space group, Γ , and aspect ratios of the unit cell can be taken as known inputs in MR computations, since they are all provided by experimental observation. And from homology modeling, it is often possible to have reliable estimates of the shape of each domain in a multi-domain protein. What remains unknown are the relative positions and orientations of these domains and the overall position and orientation of the symmetry-related copies of the proteins within the unit cell.

Once these are known, a model of the unit cell can be constructed and used as an initial

phasing model that can be combined with the X-ray diffraction data. This is, in essence, the molecular replacement approach that is now more than half a century old [75, 76]. Many powerful software packages for molecular replacement include those described in [77, 78]. Typically these perform rotation searches first followed by translation searches.

Recently full 6 degree-of-freedom (DOF) rigid-body searches and $6N$ DOF multi-rigid-body searches have been investigated [79, 80] where N is the number of domains in each molecule or complex. These methods have the appeal that the false peaks and “noise” that results when searching the rotation and translation functions separately can be reduced.

3.3 The Multi-Domain Molecular Replacement Method (MMR)

The molecular replacement (MR) method, originally developed in the 1960s [75, 81–83] is a computational method for phasing X-ray diffraction data for biomolecular structures. It has been integrated into crystallographic structure determination codes [77, 84]. Though MR has been wildly successful for single-domain proteins, significant issues arise when using MR for multi-domain proteins and complexes.

Currently two major computational paradigms exist for phasing of X-ray diffraction patterns of multi-domain proteins: (1) use existing software packages to obtain candidate peaks in the rotation function for individual domains separately, then solve for the translation function [85]; (2) attempt to morph multi-domain candidate models that contain their

CHAPTER 3.

full “6N” degrees of freedom and iteratively refine those models [79]. Both methods suffer from different aspects of the “curse of dimensionality,” which we seek to circumvent using a combination of the initial results reported in [80] and new approaches based on advanced mathematical concepts that are new to the crystallography community.

Consider a multi-domain protein or complex consisting of N rigid-bodies. If $f_i(\mathbf{x})$ denotes the density of the i^{th} body, then the density of the whole complex will be of the form $f(\mathbf{x}) = \sum_{i=1}^N f_i(g_i^{-1} \cdot \mathbf{x})$ where $g_i = (R_i, \mathbf{t}_i)$ is a rigid-body motion consisting of a rotation-translation pair and $g_i^{-1} \cdot \mathbf{x} = R_i^T(\mathbf{x} - \mathbf{t}_i)$. These motions are the unknowns in our problem.

If m identical copies of this complex are arranged symmetrically in a unit cell by symmetry operators $\gamma_j = (Q_j, \mathbf{a}_j) \in \Gamma$ (which is the group consisting of n discrete rigid-body motions that are known a priori from the crystal symmetry and geometry), an X-ray diffraction experiment provides the magnitude (without phase) of the Fourier transform of $\sum_{j=0}^{m-1} f(\gamma_j^{-1} \cdot \mathbf{x})$. In contrast, the model density for a single domain and its symmetry mates is $\sum_{j=0}^{m-1} f_i(h_i^{-1} \circ \gamma_j^{-1} \cdot \mathbf{x})$ where h_i is the candidate position and orientation. In traditional MR, the Fourier transform of the Patterson functions, $\hat{P}(g_1, \dots, g_N; \mathbf{k}) = \mathcal{F}[P(g_1, \dots, g_N; \mathbf{x})]$ and $\hat{p}_i(h_i; \mathbf{k}) = \mathcal{F}[p_i(h_i; \mathbf{x})]$, that correspond to these densities and their correlation are respectively

$$\hat{P}(g_1, \dots, g_N; \mathbf{k}) = \left| \sum_{j=0}^{m-1} \mathcal{F}[f(\gamma_j^{-1} \cdot \mathbf{x})] \right|, \quad (3.5)$$

$$\hat{p}_i(h_i; \mathbf{k}) = \left| \sum_{j=0}^{m-1} \mathcal{F}[f_i(h_i^{-1} \circ \gamma_j^{-1} \cdot \mathbf{x})] \right|, \quad (3.6)$$

$$c(h_i) = \int_{\mathbf{x} \in \mathcal{C}} P(g_1, \dots, g_N; \mathbf{x}) p_i(h_i; \mathbf{x}) d\mathbf{x} \quad (3.7)$$

where the Fourier transform \mathcal{F} converts a function of spatial position, \mathbf{x} , into a function of spatial frequency, \mathbf{k} . The real-space Pattersons themselves are obtained by applying the inverse Fourier transform. Of the quantities in (5)-(7), $\hat{P}(g_1, \dots, g_N; \mathbf{k})$ comes from the experiment (this is the multi-domain version of (2)), and $\hat{p}_i(h_i; \mathbf{k})$ and $c(h_i)$ are computed. Here \mathcal{C} is the unit cell and in MR searches the hope is that peaks in the function $c(\cdot)$ correspond to $h_i = g_i$. The difficulty is that, unlike the single domain case, in the multi-domain case P depends on many g_j 's that all interact with each other. Therefore, peaks in this rotational correlation function do not necessarily correspond to good overall matches.

3.4 Phasing by Packing

Instead of running traditional MR searches on domain orientation or full conformation, we propose to construct packing models for the multi-domain systems of interest. This will generate candidate sets of motions $\{h_1, \dots, h_N\}$ that can then be used to construct a *model* of $P(h_1, \dots, h_N; \mathbf{x})$ rather than $p_i(h_i; \mathbf{x})$. If $P(h_1, \dots, h_N; \mathbf{x})$ and $P(g_1, \dots, g_N; \mathbf{x})$ match well to each other, then that is a much stronger indication that $h_i = g_i$ than having high correlations between $p_i(h_i; \mathbf{x})$ and $P(g_1, \dots, g_N; \mathbf{x})$.

But in order for our proposed approach to work, the fraction of the total $6N$ -dimensional search space that we search must be very small. Otherwise it will be computationally expensive. In other words, we must rapidly determine “where not to look.” Preliminary results along these lines are very encouraging. We hypothesize that the combination of

CHAPTER 3.

crystal packing constraints and limitations on the range of motion between domains imposed by known motion constraints (in the case of multi-domain proteins consisting of covalently bonded rigid-domains) severely restricts the allowable motions. This leads us to believe that it will be possible to rapidly eliminate vast portions of high-dimensional configuration space based on their incompatibility with constraints, and that the enumeration of packing geometries can be performed in a computationally tractable manner.

The main contribution of our phasing approach, i.e., the approach to obtain the phase information, is that computationally expensive MR searches in continuous configuration space are replaced by a search on a relatively small discrete set of candidate packing arrangements of a multi-rigid-body model. The most important crystal packing constraint is that protein macromolecules do not collide with (or insert into) each other. With high protein-water volume ratio in crystals, they usually have to be “smartly” close packed. Since the allowable motion is severely restricted, it becomes more helpful to find a discrete candidate set to represent all the feasible packing arrangements.

We propose two methods to generate the set of candidate packing arrangements. First, it is through collision detection on a coarse grid in the configuration space. Second, we also design a Gaussian-based potential function. The candidate arrangements can be generated by minimizing this potential function that forces the model conformations to separate from each other and not overlap within the unit cell. The details of these methods are presented in the following sections.

3.4.1 Collision detection for generating candidate packing arrangements

In this approach, an ellipsoid or a combination of several ellipsoids are used to approximate the convex hull of each domain of protein structures. The reason is that the ellipsoid or the finite union of ellipsoids can be used to describe a large variety of shapes and also be expressed in simple closed-form equations. To illustrate our approach, we construct a multi-ellipsoid-shaped “rabbit” with one “face” and two “ears” as a packing model for a 3-domain structure. This rabbit-model is motivated by the 3-domain protein Lactoferrin (see Fig. 3.2). By properly modifying the size and orientation of the three ellipsoids, the rabbit model can serve as a packing model for Lactoferrin.

The most important crystal packing constraint is that protein macromolecules do not collide with (or insert into) each other. With high protein-water volume ratio in crystals, they usually have to be “smartly” close packed. Since the allowable motion is severely restricted, we can find a discrete candidate set to represent all the feasible packing arrangements. Noticing Gaussian functions have infinite tails, a Gaussian-based cost function (GCF) is constructed to evaluate the level of overlapping (or closeness) among ellipsoids with each ellipsoid represented by a Gaussian function or a mixture of Gaussian functions. The candidate packing arrangements can be obtained by minimizing the GCF to force the packing model to separate from each other and not overlap within the unit cell.

The main procedures of finding phase information using packing models can be de-

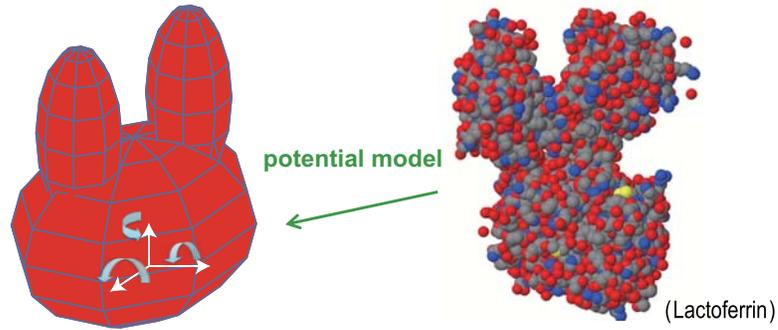


Figure 3.2: The rabbit packing model motivated by the 3-domain protein Lactoferrin.

scribed by a flowchart in Fig. 3.3. In the first step, we discretize the configuration space by a coarse grid (in 10-degree increments in this chapter), and detect collisions for the packing configuration on this grid. With the closed-form ellipsoid equation, evaluating the collisions between ellipsoids is much less computationally expensive compared to calculating $c(h_i)$ in traditional MR searches (see (3.7)). After the collision detection, we reduce the whole configuration space to a much shorter list.

In the next step, we use a Fourier-based cost function (FCF), where

$$mbxFCF(h_1, \dots, h_N) = \left[\int_{\mathbf{k} \in \Omega} (\hat{P}(g_1, \dots, g_N; \mathbf{k}) - \hat{P}(h_1, \dots, h_N; \mathbf{k}))^2 d\mathbf{k} \right]^{\frac{1}{2}}, \quad (3.8)$$

to sort these collision-free configuration from low to high. Minimizing $FCF(h_1, \dots, h_N)$ is similar to finding peaks in $c(h_i)$ except that we use a multi-domain model rather than a single-domain one. After the sorting, we keep 10 configuration with lowest FCF as a candidate list. These candidates indicate high correlations with the target structure. The FCF has the rugged surface of the configuration space, so to further improve the accuracy, a stochastic sampling method—rapidly-exploring random tree (RRT) algorithm [86] is used

to minimize the FCF around the “best candidate”. The best candidate can be first chosen as the one with the lowest FCF in the set. If its FCF cannot be reduced below a threshold value C after running the RRT, we switch the best candidate to the one with the next lowest FCF.

3.4.2 A Gaussian-based potential function for generating candidate packing arrangements

Noticing Gaussian functions have infinite tails, a Gaussian-based cost function (GCF) is constructed to evaluate the level of overlapping (or closeness) among ellipsoids with each ellipsoid represented by a Gaussian function or a mixture of Gaussian functions. The candidate packing arrangements can be obtained by minimizing the GCF to force the packing model to separate from each other and not overlap within the unit cell.

The shape of an ellipsoid can be captured by equidensity contours of a Gaussian function with the mean located at the ellipsoid center and the covariance matrix related to its semi-axis lengths. An arbitrarily oriented ellipsoid in \mathbb{R}^n can be described as

$$(\mathbf{x} - \boldsymbol{\mu})^T R^T A R (\mathbf{x} - \boldsymbol{\mu}) = 1, \quad (3.9)$$

where R is the rotation matrix, and $A = \text{diag}[1/a_1^2, 1/a_2^2, \dots, 1/a_n^2]$, with a_i denoting the semi-axis length of the ellipsoid. Compared with a Gaussian function in \mathbb{R}^n ,

$$\rho(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right), \quad (3.10)$$

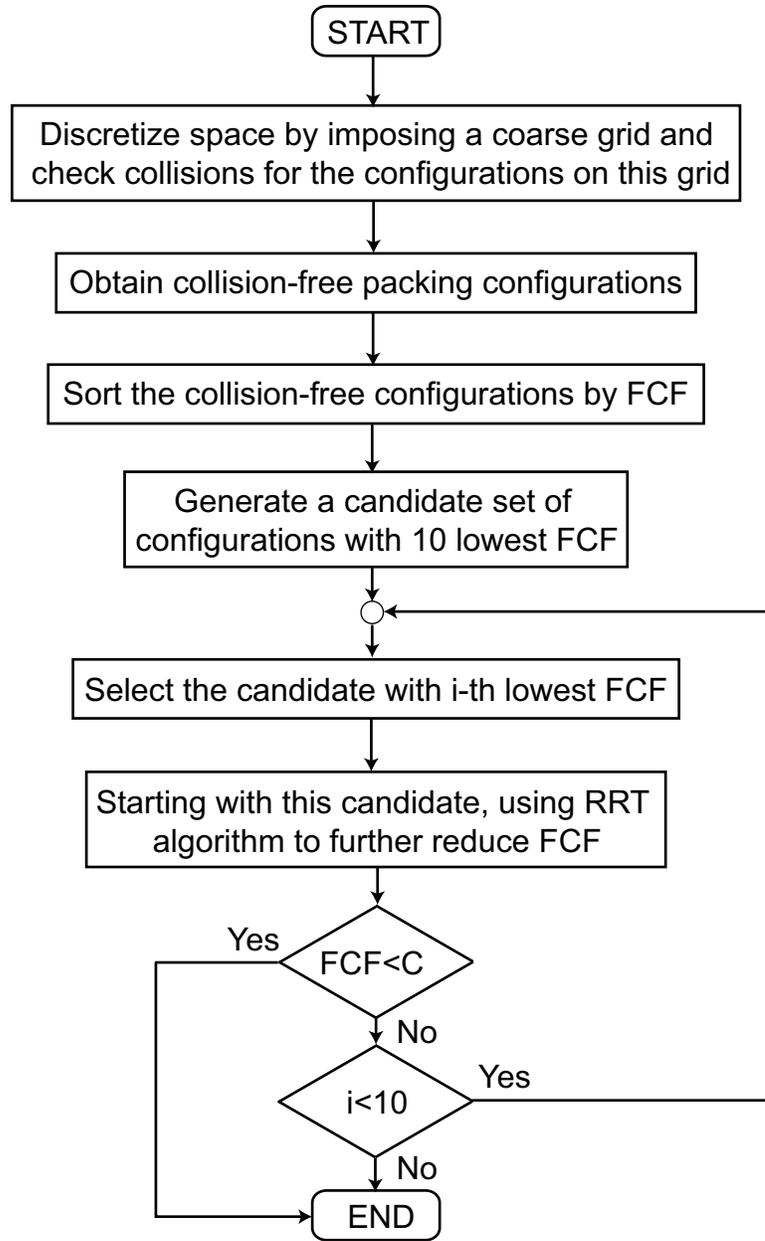


Figure 3.3: Flowchart of our approach to findin candidate phasing models. The candidate packing arrangements are generated by collision detections.

CHAPTER 3.

we can see that when $\Sigma^{-1} = R^T A R$, the equidensity contours of the Gaussian function are ellipsoids with semi-axis lengths $k \cdot a_1, k \cdot a_2, \dots, k \cdot a_n$, where $k \in \mathbb{R}_{\geq 0}$. To more accurately capture the shape of the ellipsoid with semi-axis lengths a_1, a_2, \dots, a_n , we want the Gaussian function to have high and steady value inside the ellipsoid region and a quick drop outside it. We note that it is not necessary to eliminate the tail outside the ellipsoid since the interaction among the tails can help push the ellipsoids away from each other. We use a Gaussian mixture function $\psi(\mathbf{x}; \mathbf{a}, \mathbf{b})$, i.e.,

$$\psi(\mathbf{x}; \mathbf{a}, \mathbf{b}) = \sum_{i=1}^n \frac{a_i}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{b_i}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right), \quad (3.11)$$

instead of a single Gaussian $\rho(\mathbf{x})$ to approximate an ellipsoid. In the 1D case in Fig. 3.4, with both variances $\sigma = 1$, we can see that compared to the single Gaussian $\frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$, the Gaussian mixture function with $\mathbf{a} = 0.44 \cdot [3, -1]$ and $\mathbf{b} = 1.16 \cdot [1, 3]$, i.e.,

$$\psi(x; \mathbf{a}, \mathbf{b}) = \frac{1.32}{\sqrt{2\pi}} \exp(-0.58x^2) - \frac{0.44}{\sqrt{2\pi}} \exp(-1.73x^2), \quad (3.12)$$

has a flatter top and faster decay tails.

The ellipsoid model of the i^{th} domain in a multi-domain structure under a symmetry group Γ can be approximated by $\psi((h_i^{-1} \circ \gamma_j^{-1} \cdot \mathbf{x}); \mathbf{a}, \mathbf{b})$, where h_i is rigid-body operation of the i^{th} domain and γ_j is the symmetry operator in the symmetry group Γ . Therefore we define the GCF as

$$\text{GCF}(h_1, \dots, h_N) \triangleq \int_{\mathbb{R}^n} \left[\sum_{j=0}^{m-1} \sum_{i=1}^N \psi((h_i^{-1} \circ \gamma_j^{-1} \cdot \mathbf{x}), \mathbf{a}, \mathbf{b}) \right]^2 d\mathbf{x}. \quad (3.13)$$

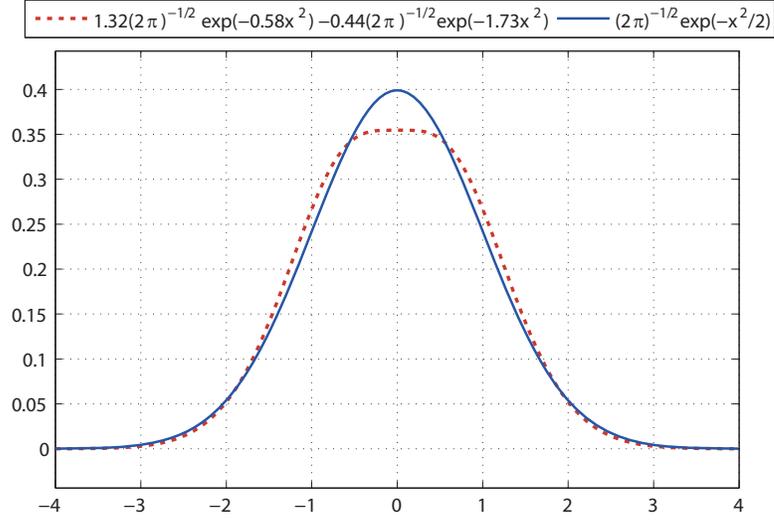


Figure 3.4: The comparison between a single Gaussian with a mixture of Gaussians.

An advantage of Gaussian functions is that the integration of quadratic terms over \mathbb{R}^n has a closed-form expression. We derive it as follows,

$$\begin{aligned}
 & \int_{\mathbb{R}^n} \rho_1(\mathbf{x}; \mu_1, \Sigma_1) \rho_2(\mathbf{x}; \mu_2; \Sigma_2) d\mathbf{x} & (3.14) \\
 &= \int_{\mathbb{R}^n} (2\pi)^{-n/2} |\det \Sigma_1|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_1)^T \Sigma_1^{-1}(\mathbf{x} - \mu_1)\right) \\
 & \quad (2\pi)^{-n/2} |\det \Sigma_2|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_2)^T \Sigma_2^{-1}(\mathbf{x} - \mu_2)\right) d\mathbf{x} \\
 &= (2\pi)^{-n} |\det \Sigma_1 \det \Sigma_2|^{-1/2} \int_{\mathbb{R}^n} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_1)^T \Sigma_1^{-1}(\mathbf{x} - \mu_1)\right) \\
 & \quad - \frac{1}{2}(\mathbf{x} - \mu_2)^T \Sigma_2^{-1}(\mathbf{x} - \mu_2)\right) d\mathbf{x}.
 \end{aligned}$$

$$\begin{aligned}
 & \text{Since } \int_{\mathbb{R}^n} \exp\left(-\frac{1}{2}\mathbf{x}^T M \mathbf{x} - m^T \mathbf{x} - C\right) d\mathbf{x} & (3.15) \\
 &= (2\pi)^{n/2} |\det M|^{-1/2} \exp\left(\frac{1}{2}m^T M^{-1}m - C\right),
 \end{aligned}$$

(3.14) can be rewritten in a closed-form as

$$\begin{aligned}
& \int_{\mathbb{R}^n} \rho_1(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \rho_2(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) d\mathbf{x} & (3.16) \\
& = (2\pi)^{-n/2} |\det \boldsymbol{\Sigma}_1 \det \boldsymbol{\Sigma}_2 \det(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})|^{-1/2} \\
& \quad \exp\left(\frac{1}{2}(\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\mu}_2^T \boldsymbol{\Sigma}_2^{-1})(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})(\boldsymbol{\Sigma}_1^{-T} \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-T} \boldsymbol{\mu}_2) \right. \\
& \quad \left. - \frac{1}{2}(\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^T \boldsymbol{\Sigma}_2^{-1} \boldsymbol{\mu}_2)\right).
\end{aligned}$$

The closed-form expression of the GCF can be easily derived from (3.16).

The main procedures of generating candidate phasing models by packing can be described by a flowchart in Fig. 3.5. In the first step, we discretize the configuration space by a coarse grid (much coarser than by the method of collision detections), and find the configuration with the smallest GCF value inside each “configuration cell” defined by the grid. The collision-free ones of these configurations form the candidate set of packing arrangements. This discrete candidate set reduces the whole configuration space to a much shorter list. We note that with a closed-form expression, minimizing the GCF is less computationally expensive compared to calculating $c(h_i)$ in traditional MR searches (see (3.7)). The rest of procedures are the same as the method of collision detection (see Sec. 3.4.1).

3.5 Numerical Results

In this section, we demonstrate the packing approach with the rabbit model in both the planar and spatial cases. We note that the Gaussian-based method (Sec. 3.4.2) is more favorable for planar cases because with a smaller set of local minima, each local minimum

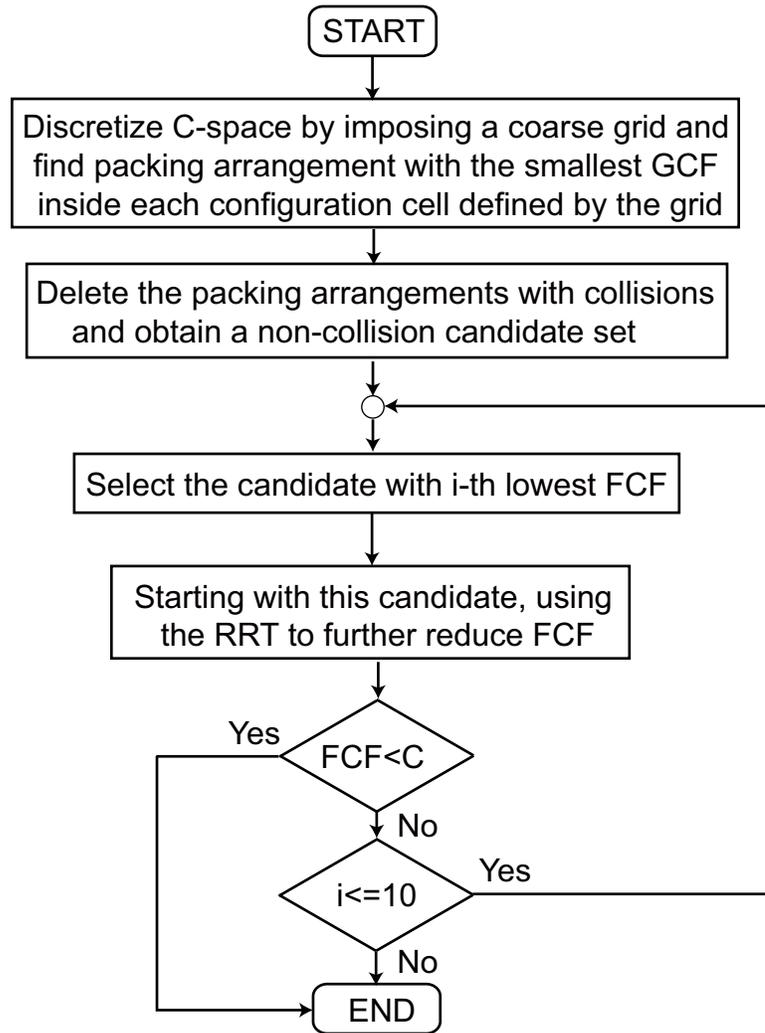


Figure 3.5: Flowchart of our approach to finding candidate phasing models. The candidate packing arrangements are generated by minimizing the Gaussian-based potential function.

CHAPTER 3.

can be confined within a “configuration cell” defined by a coarser grid. But for spatial cases, the local minima of GCF are in a more noisy landscape, which makes the optimization process more difficult. It turns out to be faster to directly detect collisions on the grid points. So in the simulation results, we only demonstrate this method in planar examples in the context of $p1$ wallpaper symmetry. For the spatial cases, we apply the method of collision detections, and it is demonstrated the simplest space-group symmetry, $P1$, and the most common space-group symmetry in protein structure, $P2_12_12_1$.¹

3.5.1 A planar example: $p1$ symmetry

In our planar example, we use ellipses to represent both the domains of the model and target structures. All the angular parameters of the target structure are treated as being unknown, and the only known information that we have is the magnitude of the Fourier transform of the electron density function $\hat{P}(g_1, \dots, g_N; \mathbf{k})$. Our goal is to find the closest model configuration $\{h_1, \dots, h_N\}$ with respect to the target structure $\{g_1, \dots, g_N\}$. To illustrate our approach, a multi-ellipse-shaped “rabbit” with one “face” and two “ears” is constructed as a packing model for a 3-domain structure in $p1$ symmetry. Since translations have no impact on the packing result in $p1$ symmetry, the rabbit model has 3 DOF—the rotations of the face, θ_1 and two ears, θ_2 and θ_3 (see the dimensions and ranges of motion in Tab. 3.1).

¹Conventionally, the lowercase p is used to represent the planar wallpaper symmetries, while the capital P is for the spatial crystallographic symmetries.

CHAPTER 3.

To build the Gaussian mixture function in the 2D planar case, we use the same ratio for the two Gaussian functions as the 1D case in (3.12), i.e., $a = m_a \cdot [3, -1]$ and $b = m_b \cdot [1, 3]$. Here, m_a and m_b are two positive scalars. m_b^* — the optimal value of m_b , is chosen to “stretch or shrink” the Gaussian mixture function so that it can “best” represent the define ellipse. After that, m_a^* is calculated to normalize the Gaussian mixture function with m_b^* . More specificall , we defin m_b^* as

$$m_b^* = \arg \max_{m_b} |S_{\text{cand.}}(m_b)|, \quad (3.17)$$

where $S_{\text{cand.}}$ represents the non-collision candidate set, generated by obtaining the packing arrangements with the smallest GCF value inside each configuratio box define by the grid. We note that although the packing arrangements with lower GCF values are less like to be in collision, a local minimum of GCF cannot always guarantee a collision-free packing arrangement. So before including all the packing arrangements with lowest GCF values into the set $S_{\text{cand.}}$, we rule out the ones that are in collision first. $|S_{\text{cand.}}(m_b)|$ denotes the number of non-collision candidates in this set. With the optimal m_b , the GCF forces the packing models to separate from each other to the greatest extent, and the size of the non-collision candidate set is therefore maximized. Fig. 3.6 shows the size of the non-collision candidate set $|S_{\text{cand.}}(m_b)|$ with different m_b values under 3 different define grids (in 30-, 40- and 60-degree increments). We can see when $m_b = 0.2$, $|S_{\text{cand.}}(m_b)|$ has the highest value, and the peak is independent of how we defin the grid. In the experiment, we use the 30-degree grid, and 48 non-collision candidates can be found. With $m_b = 0.2$, we compare the contours of the Gaussian mixture function with the rabbit shape in Fig.

CHAPTER 3.

3.7, and we can see that it fits the shape of the rabbit model well. Also in Fig. 3.8, we compare collision checking results with GCF values in the θ_1 - θ_2 plane with fixed $\theta_3 = -90$ degrees. It is shown that all non-collision configurations are located in the low GCF value regions, which demonstrate that by minimizing the GCF, the ellipses are less likely to have overlapping.

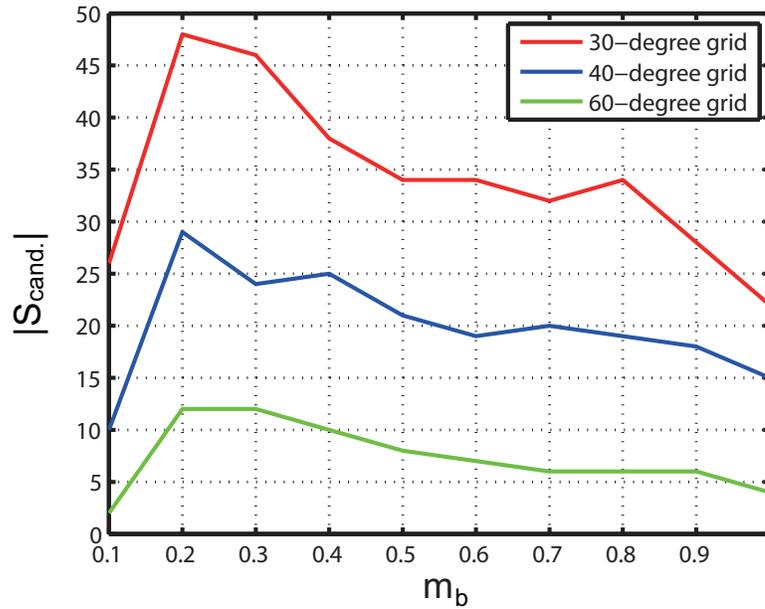


Figure 3.6: The size of the non-collision candidate set with different m_b values under 3 different define grids (in 30-, 40- and 60-degree increments, respectively). $|S_{\text{cand.}}(m_b)|$ denotes the number of non-collision candidates in this set.

An example of packing results with the target structure randomly sampled in space is illustrated in Figs. 3.9 and 3.10. After generating the candidate set by minimizing the FCF, and sorting these candidates by the FCF from high to low (the top three are listed in Fig. 3.9), the best candidate in the set (Candidate 1 in Fig. 3.9) shows 1.50, 17.81 and 10.97 degrees of the error in θ_1 , θ_2 and θ_3 , respectively. After running the RRT in the neighbor-

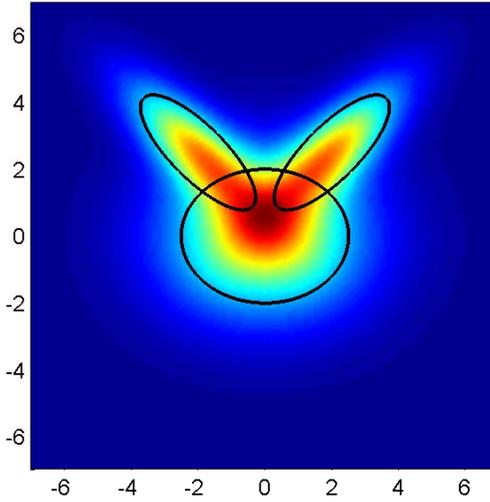


Figure 3.7: The comparison of the rabbit shape with the contours of the Gaussian mixture function ($m_b = 0.2$).

hood of this candidate's configuration these errors are further reduced to only 0.79, 2.14 and 0.19 degrees respectively, less than 1.2 % of the total rotation range. Tab. 3.2 shows 10 different numerical trials and the mean absolute errors (MAE), $\text{mean}\{\Delta\theta_1, \Delta\theta_2, \Delta\theta_3\}$, are all below 3 degrees.

3.5.2 Spatial examples: $P1$ and $P2_12_12_1$ symmetries

In our spatial examples, we also construct a rabbit model with three ellipsoids (one for the face, the other two for the two ears) representing a three domain structure. In the numerical experiments, the same packing model is used each time to construct target structures. All of the angular parameters of the target structure are unknown for the algorithm, and the only known information is the magnitude of the Fourier transform of the electron density

CHAPTER 3.

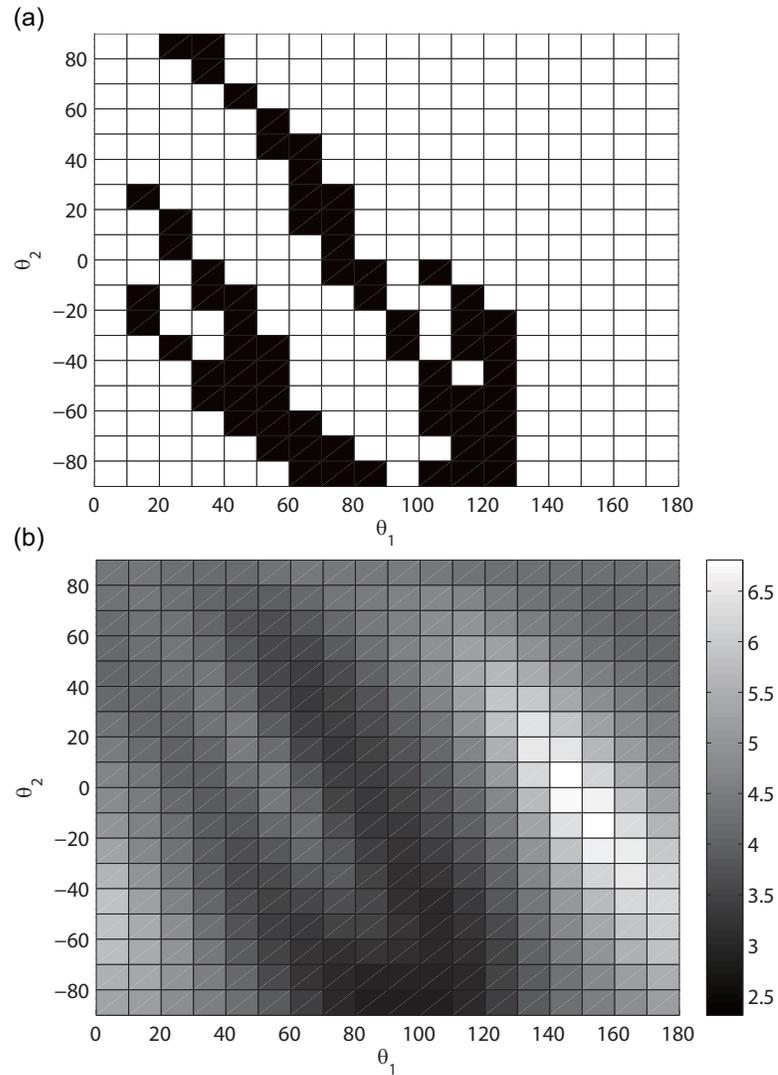


Figure 3.8: The comparison of (a) collision checking results with (b) GCF values (with $m_b = 0.2$) in the θ_1 - θ_2 plane (with $\theta_3 = -90$ degrees). In (a), black pixels represent the collision-free configuration and white ones are in collision. In (b), the pixels with darker colors represent the configuration with lower GCF values, and vice versa.

CHAPTER 3.

Table 3.1: The dimensions and ranges of motion of the rabbit packing model.

Dimensions	size of the unit cell	9×6.75
	semi-axis lengths of the face	2; 2.5
	semi-axis lengths of the ears	2.3; 0.92
Range of rotation	face: θ_1 (deg)	$0 \sim 180$
	ears: θ_2, θ_3 (deg)	$-90 \sim 90$

Table 3.2: 10 Numerical trials for the planar crystal packing example. The angles are all in degrees.

Trial	Target Structures			Best Cand.'s Errors			Final Errors		
	θ_1	θ_2	θ_3	$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$	$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$
1	100.82	-72.21	-3.03	0.52	17.79	10.97	0.79	2.14	0.19
2	42.29	64.37	-69.25	0.22	4.37	9.25	0.67	0.08	2.08
3	136.67	-68.70	-67.33	16.67	29.33	12.65	1.09	1.16	1.06
4	114.21	-63.46	-51.42	5.79	17.57	3.46	2.22	1.25	1.72
5	54.83	-49.51	-70.41	7.02	10.49	10.41	1.00	0.86	1.04
6	159.67	47.67	-2.65	14.3	20.9	17.45	1.08	3.15	3.08
7	101.63	-67.65	12.06	12.35	20.76	18.99	2.09	2.55	1.02
8	113.89	-73.69	30.76	6.11	16.31	8.19	1.17	1.11	1.06
9	66.41	27.29	-76.94	6.41	11.66	13.06	3.21	3.49	1.55
10	97.19	-1.59	-86.46	22.81	37.78	6.48	2.83	1.30	2.93

CHAPTER 3.

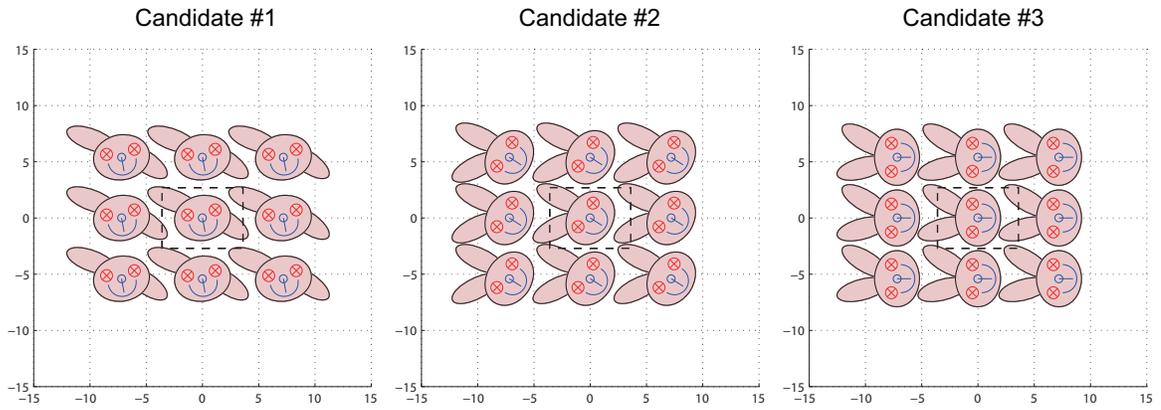


Figure 3.9: 3 candidate packing arrangements for the planar crystal packing example.

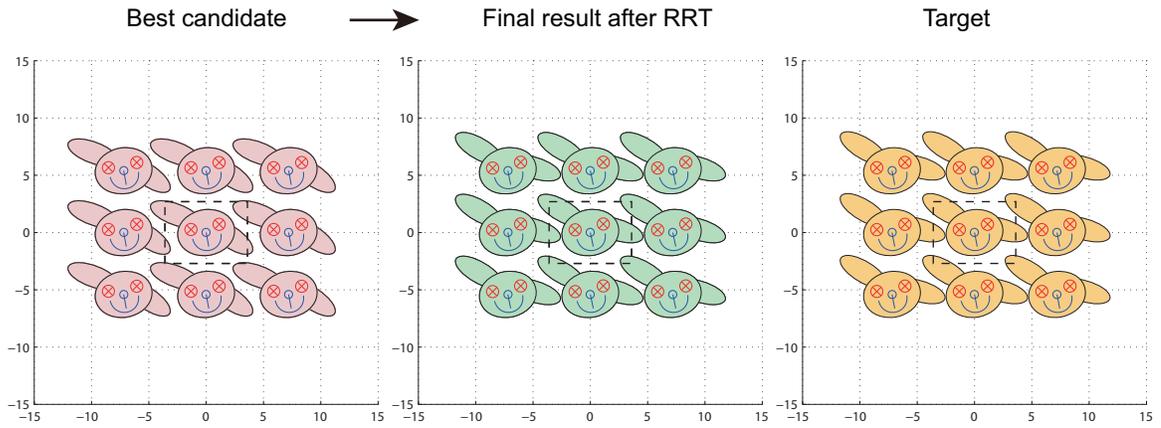


Figure 3.10: The planar crystal packing example with the target structure randomly sampled in the space.

CHAPTER 3.

function $\hat{P}(g_1, \dots, g_N; \mathbf{k})$. Our goal is to find the closest model configuration $\{h_1, \dots, h_N\}$ with respect to the target structure $\{g_1, \dots, g_N\}$. The examples are shown in the context of $P1$ and $P2_12_12_1$ symmetries.

3.5.2.1 Results in $P1$ symmetry

The rabbit model has 7 degrees of freedom (DOF) —roll (α_1), pitch (β_1) and yaw (γ_1) of the face and rolls (α_2, α_3) and pitches (β_2, β_3) of the two axisymmetric ears (Fig. 3.15). The most important constraint of the motion is that the rabbits cannot collide with (or insert into) each other. With 50% volume ratio between the packing model and the unit cell (see the dimensions in Tab. 3.5), there is not much free room to move for the packing model. So the rabbits have to be “smartly” close packed in the configuration space to avoid collision, as most protein molecules are in real crystals. Fig. 3.11 shows examples of packing configuration with and without collisions using our packing model in $P1$ symmetry, and the yellow part in (a) shows the collision areas. Also, some constraints on the motion between domains are imposed (see the ranges of motion for each DOF in Tab. 3.3).

To evaluate the packing results, three different errors— E_h , MAE, and E_{max} are defined as,

$$E_h = \sum_i^N \|g_i - h_i\|_W, \quad (3.18)$$

$$E_{max} = \max\{\Delta\alpha_1, \Delta\beta_1, \Delta\gamma_1, \Delta\alpha_2, \Delta\beta_2, \Delta\alpha_3, \Delta\beta_3\},$$

$$\text{MAE} = \text{mean}\{|\Delta\alpha_1|, |\Delta\beta_1|, |\Delta\gamma_1|, |\Delta\alpha_2|, |\Delta\beta_2|, |\Delta\alpha_3|, |\Delta\beta_3|\},$$

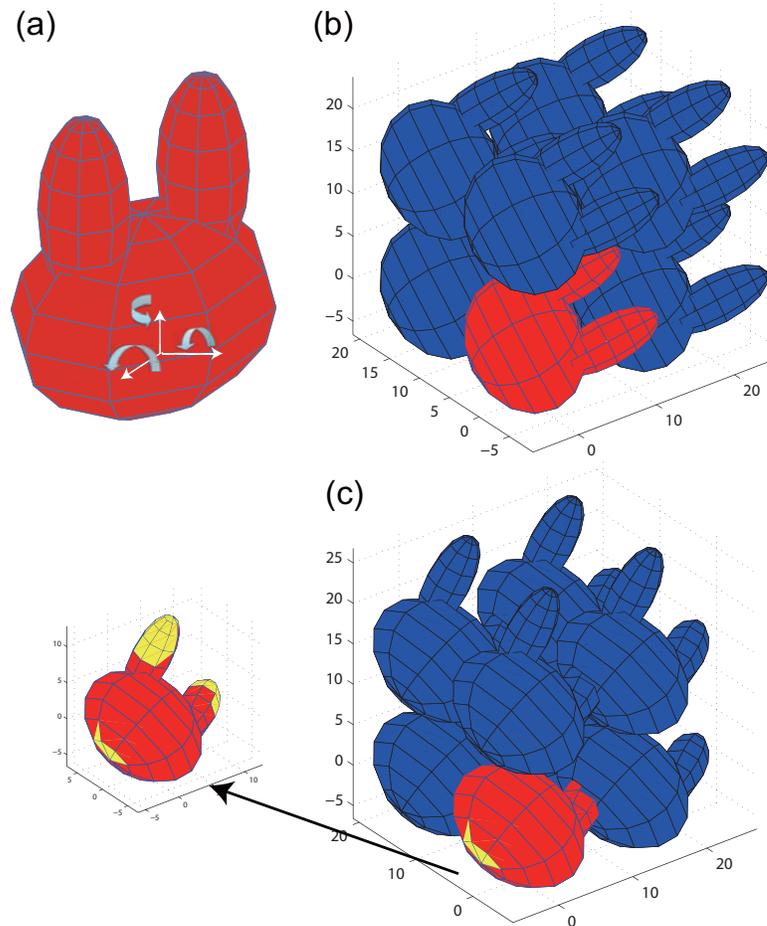


Figure 3.11: (a) Illustration of 3 degrees of freedom in the packing model. Examples of packing configuration (b) without collisions (c) with collisions (indicated in yellow).

CHAPTER 3.

where E_h is the error metric of motion $\{h_1, \dots, h_N\}$ relative to (g_1, \dots, g_N) and W is the weight matrix $\begin{pmatrix} J & \mathbf{0} \\ \mathbf{0}^T & M \end{pmatrix}$ as in [87]. Since the examples in this section have symmetry group $\Gamma=P1$, and since there is no translation involved in P1 symmetry, g_i and h_i reduce to pure translations and W reduces to $J = \int_V \mathbf{x}\mathbf{x}^T \rho(\mathbf{x})dV$. For ellipsoids, $J = \text{diag}[Mr_x^2/5, Mr_y^2/5, Mr_z^2/5]$, where M is the mass and r_x, r_y and r_z are the semi-axis lengths. Here $\rho(x)$ is taken to be 1 and semi-axis lengths are reported in Tab. 3.3. We note that the absolute value of E_h depends on the mass of the model. Also, E_{max} and MAE are maximum error and mean absolute error of the angle parameters, respectively.

To demonstrate the proposed approach, the angular parameters of target structures are generated in two ways: 1) chosen from the grid; 2) randomly sampled in the configuration space. We note that all the target structures should be collision free due to the physical constraints in the real world. In case 1 (see the example in Fig. 3.12), the best candidate in the set is identical to the target structure, with three zero errors and zero FCF. When the target structure is randomly generated in the configuration space, as in case 2 (see the examples in Fig. 3.13), we can see that the set of candidates (Tab. 3.4) show similar conformations as the target structure and the best candidate in the set (Cand.1) has only 3.9 degrees of MAE and 7.1 degrees of E_{max} . After running the RRT stochastic search for 30 steps, MAE is further reduced to 2.5 degrees and E_h is also decreased by 50%. Fig. 3.14 shows the trends of errors before and after applying the RRT. The plot is generated by the results of 20 trials. In the figure we can see both E_h and MAE go down as the RRT is running. We note that the reason for the slight increase in E_{max} may be caused by the

CHAPTER 3.

different importance of the face and ears in FCF. The RRT places more weight on putting the face on the right position while the ears could be off alignment to a small extent.

Table 3.3: The dimensions and ranges of motion of the packing model in $P1$ symmetry.

Dimensions	size of the unit cell	$14 \times 14 \times 14$
	semi-axis lengths of the face	8; 6; 6
	semi-axis lengths of the ears	2.5; 2.5; 6
Face	range of roll (deg)	$0 \sim 90$
	range of pitch (deg)	$0 \sim 90$
	range of yaw (deg)	$0 \sim 90$
Ears	range of roll (deg)	$-30 \sim 30$
	range of pitch (deg)	$-30 \sim 30$

3.5.2.2 Results in $P2_12_12_1$ symmetry

In $P2_12_12_1$ symmetry, the rabbit model has 10 DOF—roll (α_1), pitch (β_1) and yaw (γ_1) of the face, rolls (α_2 , α_3) and pitches (β_2 , β_3) of the two axis-symmetric ears and translation in x -, y - and z - directions (P_x , P_y , and P_z). (see Fig. 3.15). Fig. 3.16 shows examples of packing configuration with and without collisions using our packing model in $P2_12_12_1$ symmetry, and the yellow part in (a) shows collision areas. Also, some constraints on the motion between domains are imposed (see the ranges of motion for each DOF in Tab. 3.5).

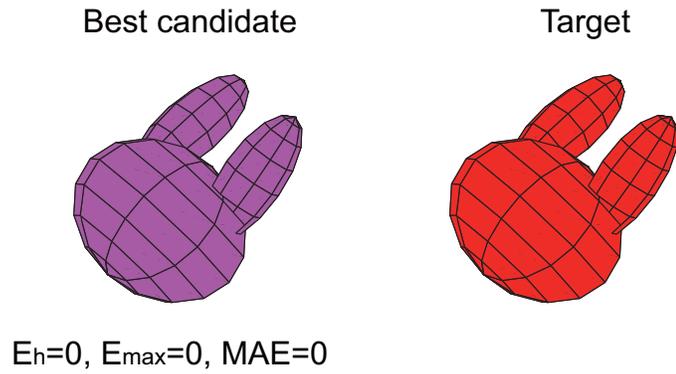


Figure 3.12: An example of packing results with the target structure on the grid in $P1$ symmetry.

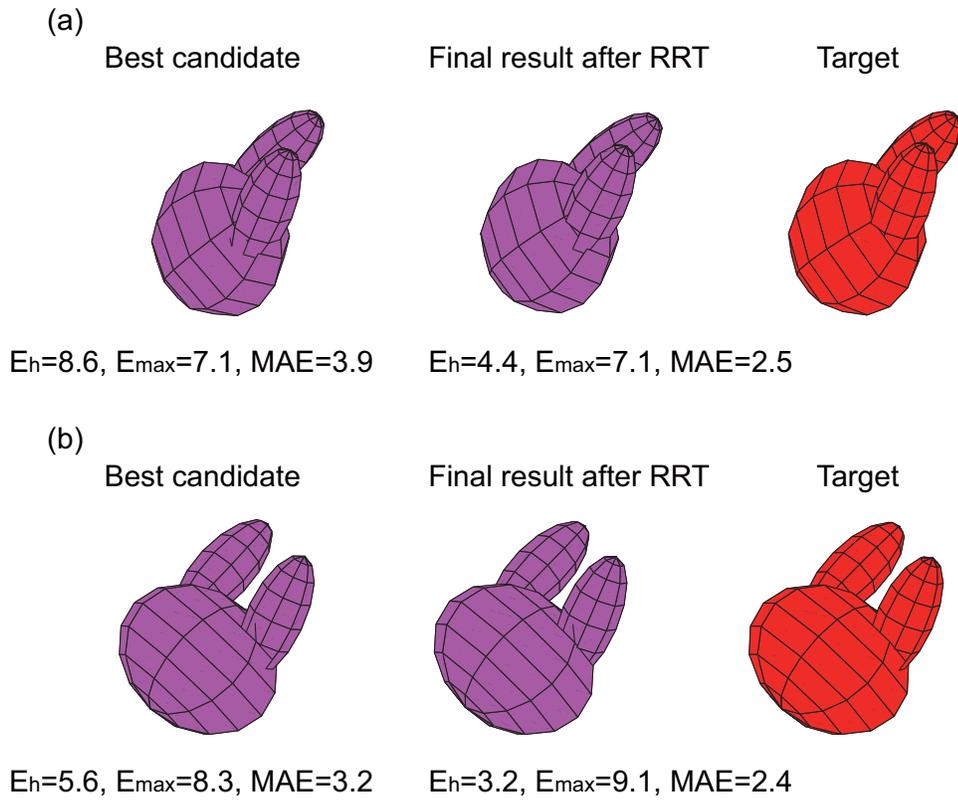


Figure 3.13: An example of packing results with the target structure randomly sampled in the space in $P1$ symmetry.

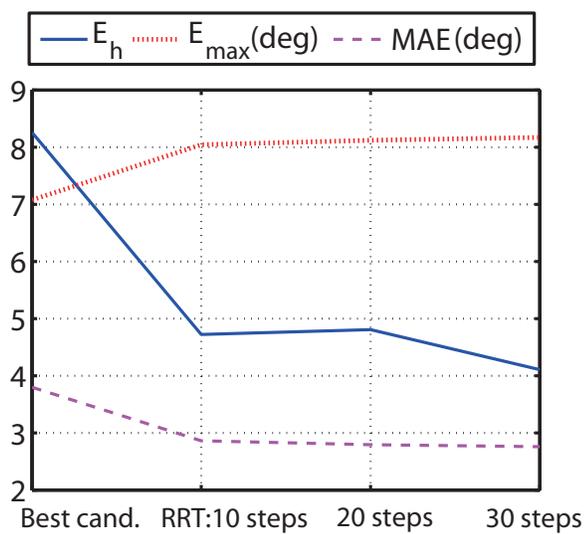


Figure 3.14: The trend of errors before and after applying the RRT in $P1$ symmetry.

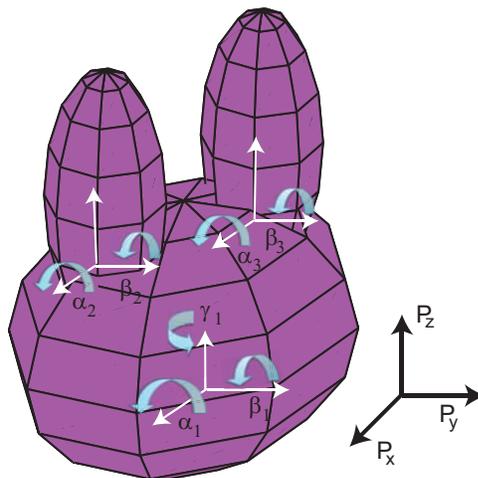


Figure 3.15: Illustration of 10 degrees of freedom in the packing model.

CHAPTER 3.

Table 3.4: The motions of the target structure and the set of candidates in the example of Fig. 3.13 in $P2_12_12_1$ symmetry. The angles are all in degrees.

	α_1	β_1	γ	α_2	β_2	α_3	β_3
Target	46.3	50.2	11.0	-29.6	-29.9	-0.09	-12.1
Cand.1	45	45	15	-25	-25	-5	-5
Cand.2	45	45	15	-25	-25	5	-5
Cand.3	45	45	15	-25	-25	15	-5
Cand.4	55	55	5	-25	-25	-15	-25
Cand.5	55	55	5	-25	-25	-25	-25
Cand.6	35	45	15	-25	-25	-5	-5
Cand.7	35	45	15	-25	-25	5	-5
Cand.8	35	45	15	-25	-25	15	-5
Cand.9	35	45	15	-25	-25	25	-5
Cand.10	35	45	15	-25	-25	-15	-5

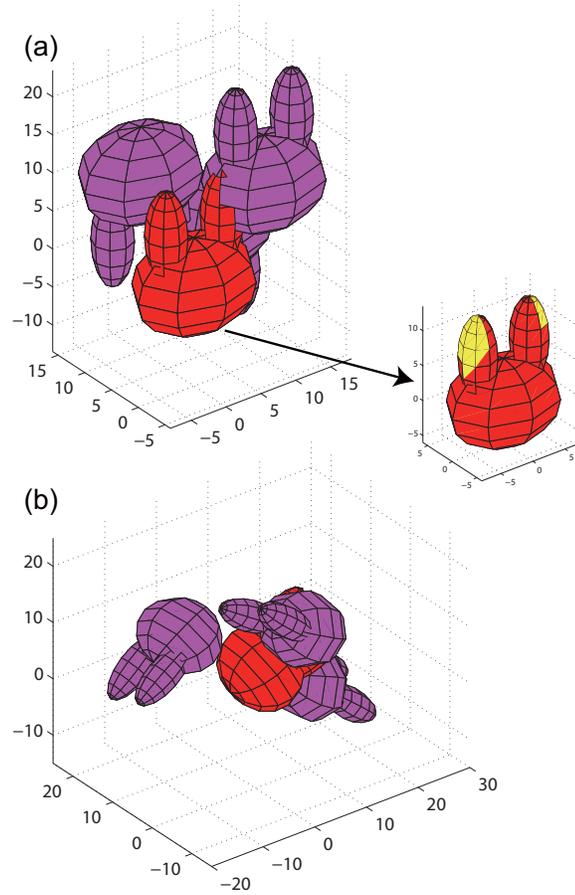


Figure 3.16: The examples of packing configuration (a) with collisions and (b) without collisions.

CHAPTER 3.

Table 3.5: The dimensions and ranges of motion of the packing model in $P2_12_12_1$ symmetry.

Dimensions	size of the unit cell	$20 \times 20 \times 20$
	semi-axis lengths of the face	8; 6; 6
	semi-axis lengths of the ears	2.5; 2.5; 6
Face	range of roll (deg)	$0 \sim 90$
	range of pitch (deg)	$0 \sim 90$
	range of yaw (deg)	$0 \sim 90$
Ears	range of roll (deg)	$-30 \sim 30$
	range of pitch (deg)	$-30 \sim 30$
Translation	x -axis	$0 \sim 4$
	y -axis	$0 \sim 4$
	z -axis	$0 \sim 3$

CHAPTER 3.

We use the same procedure to demonstrate this approach in the experiment as *P1* Symmetry. Similarly, to evaluate the packing results, three errors— E_h , MAE_1 , and MAE_2 , are define as,

$$E_h = \sum_{i=1}^3 \|g_i - h_i\|_W, \quad (3.19)$$

$$MAE_1 = \text{mean}\{|\Delta\alpha_1|, |\Delta\beta_1|, |\Delta\gamma_1|, |\Delta\alpha_2|, |\Delta\beta_2|, |\Delta\alpha_3|, |\Delta\beta_3|\},$$

$$MAE_2 = \text{mean}\{|P_x|, |P_y|, |P_z|\}.$$

where E_h is the same error used for *P1* space symmetry. MAE_1 , and MAE_2 are define as the mean absolute errors for rotations and translations, respectively.

The target structures are also generated in two ways: 1) strictly on the grid; 2) randomly sampled in the space. In case 1 (see the example in Fig. 3.17), the best candidate in the set is identical to the target structure, with three zero errors and zero FCF. When the target structure is randomly generated in space, as in case 2 (see the example in Fig. 3.18 and Tab. 3.6), we can see that the set of candidates show similar conformations as the target structure and the best candidate in the set (Cand.1) has only 4.07 degrees of MAE_1 and 0.1277 of MAE_2 . After using the RRT to search stochastically in the neighborhood of the best candidates for 20 steps, MAE_1 and MAE_2 are further reduced to 3.65 degrees and 0.079, respectively and E_h is also decreased by 30%. Fig. 3.19 shows the trends of errors before and after applying the RRT. The plot is generated by the results of 10 trials.

CHAPTER 3.

Table 3.6: The motions of the target structure and the set of candidates in the example of Fig. 3.18 in $P2_12_12_1$ symmetry. The angles are all in degrees.

	α_1	β_1	γ_1	α_2	β_2	α_3	β_3	P_x	P_y	P_z
Target	88.21	60.99	23.21	26.29	-2.14	-16.07	-19.21	3.79	3.84	2.97
Cand.1	85	65	25	25	-15	5	-15	4	4	3
Cand.2	85	65	25	25	-15	-15	-5	4	4	3
Cand.3	85	65	25	25	-15	-5	-15	4	4	3
Cand.4	85	65	25	25	-15	-25	-15	4	4	3
Cand.5	85	65	25	-25	-15	15	-15	4	4	3
Cand.6	85	65	25	-25	-15	25	-15	4	4	3
Cand.7	85	65	25	-25	-15	5	-5	4	4	3
Cand.8	85	65	25	-25	-15	15	-5	4	4	3
Cand.9	85	65	25	-25	-15	25	-5	4	4	3
Cand.10	85	65	25	-25	-15	-25	-5	4	4	3

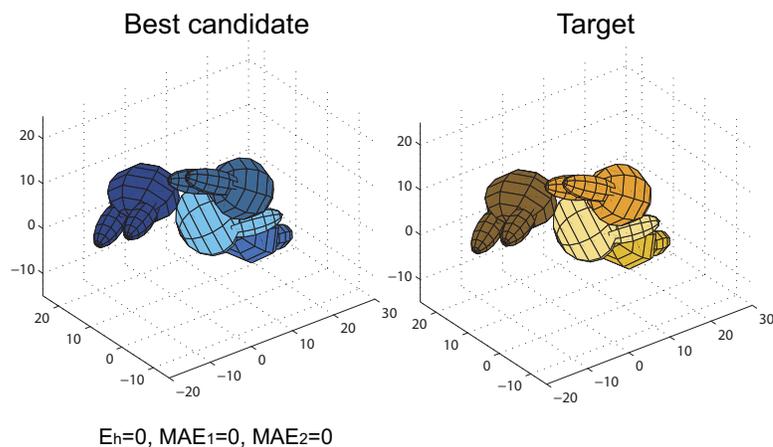


Figure 3.17: An example of packing results with the target structure on the grid in $P2_12_12_1$ symmetry.

3.6 Conclusion

Macromolecular crystallography has been the traditional workhorse for determining structural models in the field of biophysics. Within macromolecular crystallography, the molecular replacement method has been a highly successful method for providing phasing models to combine with experimental information to obtain 3D models. In this chapter we demonstrate that an alternative to molecular replacement, called “phasing by packing” is promising for multi-rigid-domain structures. Numerical results illustrate the potential of this method in the context of both the simplest space-group symmetry, $P1$, and the most common space-group symmetry in protein structure, $P2_12_12_1$. Part of the work in this chapter is published in [9–12].

CHAPTER 3.

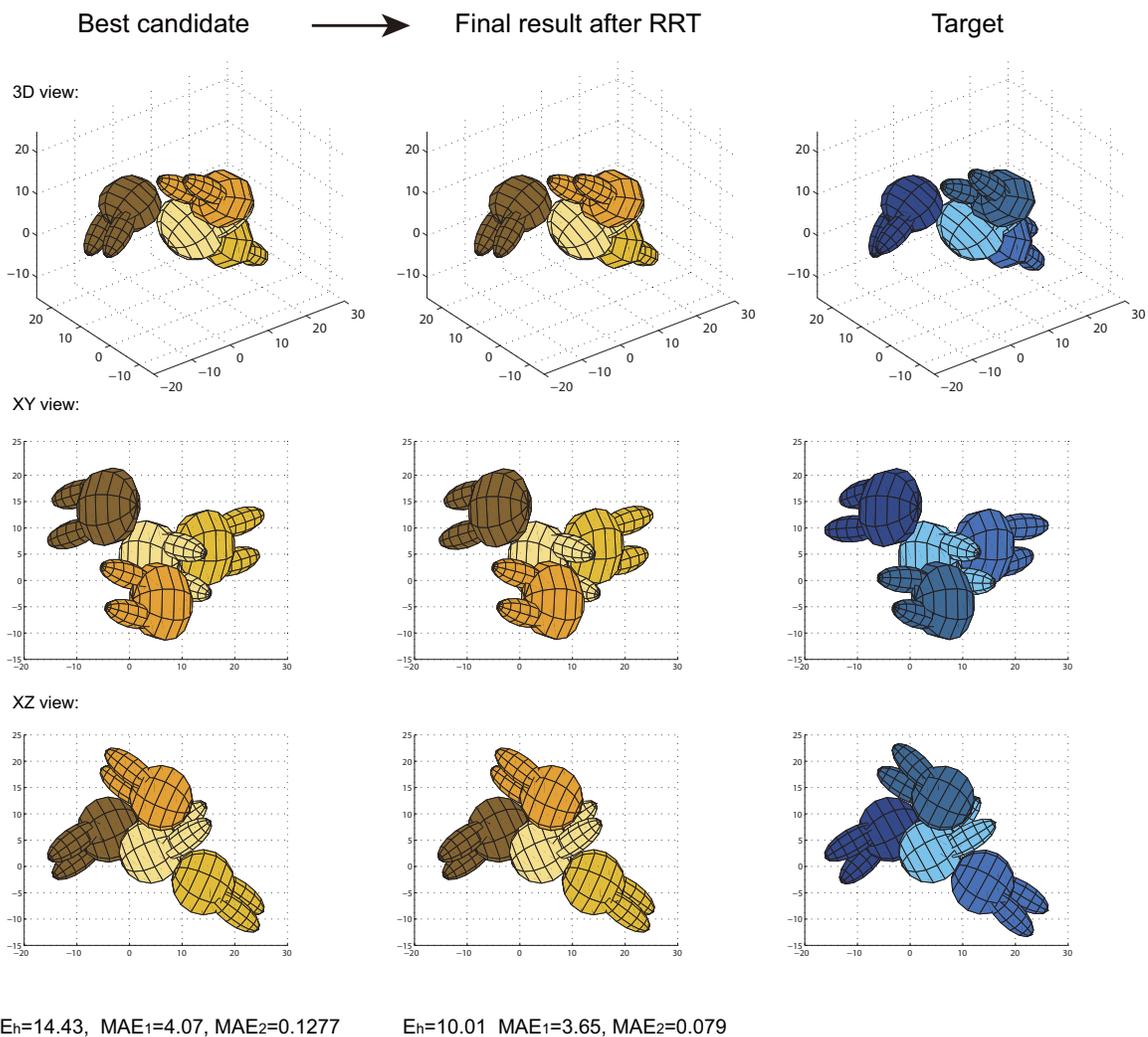


Figure 3.18: An example of packing results with the target structure randomly sampled in the space in $P2_12_12_1$ symmetry.

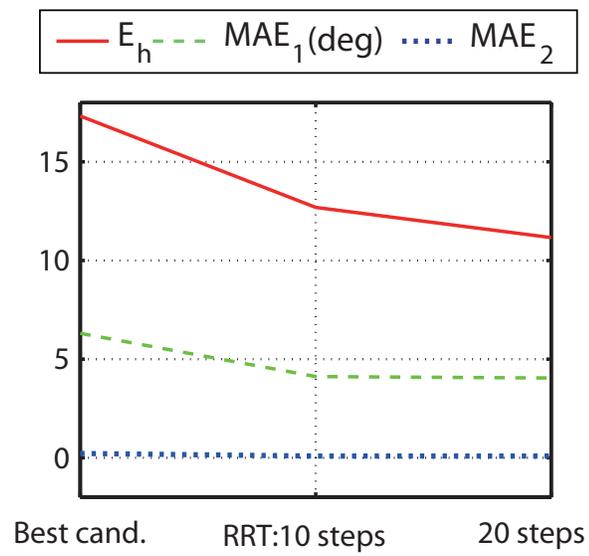


Figure 3.19: The trend of errors before and after applying the RRT in $P2_12_12_1$ symmetry.

Chapter 4

An Almost-Uniform Sampling of Rigid-Body Rotations and Motions

Uniform sampling of rotations and motions is desirable in many applications such as computational structural biology, robot motion planning and computer graphics. In this chapter, we propose a new method for sampling the rotation group and the special Euclidean group. This method involves decomposing them into identical Voronoi cells centered on the elements in some of their discrete subgroups. This approach can achieve almost-uniform sampling at any level of resolution, without having to store large numbers of coordinates, and without requiring sophisticated data structures.

The rotation group and special Euclidean group both contain discrete subgroups. In the case of the rotation group, these subgroups are the chiral point groups, and in the case of the special Euclidean group, the discrete subgroups are the chiral crystallographic space

groups. Our approach is based on sampling on these groups by partitioning them using single- and further double-coset decompositions. Fundamental domains associated with the coset- and double-coset decompositions can be defined as Voronoi cells in the original groups. Division of these groups into Voronoi cells facilitates almost-uniform sampling. We analyze the structures of the fundamental domains of these Lie groups and also demonstrate this approach in the context of a crystal packing problem.

4.1 Introduction

The group of rotations in three-dimensional space, $SO(3)$, and the groups of rigid-body motions of the plane and space, $SE(2)$ and $SE(3)$, are ubiquitous in the field of estimation and control [88–97], robotics [98,99], and computer vision [100,101]. These are Lie groups that contain discrete subgroups. In the case of $SO(3)$, the discrete subgroups are the chiral point groups, which are finite¹. Of these, we shall only be concerned with the groups of rotational symmetries of the Platonic solids since they fill $SO(3)$ more uniformly than other finite subgroups. In the case of $SE(2)$, the discrete subgroups of interest are the five chiral wallpaper groups consisting of lattice translations in the x-y plane and either no rotation, or rotation around the z axis by $2\pi/n$ radians where $n = 2, 3, 4$ or 6 .

In many application areas ranging from robot motion planning to computational structural biology, the issue of how to uniformly sample rotations arises. The problem of sam-

¹A chiral symmetry group refers to one that preserves orientation, or equivalently, the right-handedness of coordinate systems. A point group is one for which the action on Euclidean space keeps a point fixed.

CHAPTER 4.

pling uniformly at random is a very different problem than constructing schemes to uniformly sample deterministically. The former can be achieved by using the standard inverse-function technique, for example, the Euler angle parameterization, or by using one of the many other techniques described in [102–104]. Random uniform sampling on the rotation group, $SO(3)$, is much easier than deterministic uniform sampling, the latter of which does not even have an exact solution (except for the very coarse samplings corresponding to the rotational symmetry operations of the Platonic solids). This is because random sampling only depends on the Jacobian determinant, $|J(\mathbf{q})|$ (define in (2.22)), when a particular parameterization of rotation matrices, $R(\mathbf{q})$, is used. Here, either J_L or J_R define in Sec. 2.2 can be used. In contrast, a measure of equal spacing depends on the invariant distance that is used. A distortion measure then can be constructed that measures how different the metric tensor is from the identity matrix as

$$C(\mathbf{q}) = \frac{1}{\sqrt{3}} \|G(\mathbf{q}) - \mathbb{I}\|. \quad (4.1)$$

Here $G(\mathbf{q}) = J^T(\mathbf{q})J(\mathbf{q})$, $\|\cdot\|$ denotes the Frobenius norm, and $\sqrt{3} = \|\mathbb{I}\|$ is used as a normalizing factor. This is the distortion measure on the intrinsic parameterization. It is known that it is not possible to construct perfectly uniform finely spaced samples in the sense of having $C(\mathbf{q}) = 0$ for all possible values of \mathbf{q} for $R(\mathbf{q})$ to densely fill $SO(3)$. The next natural question to then ask is “how good of a fine sampling of $SO(3)$ is achievable?”

The issue of “as uniform as possible” sampling for $SO(3)$ is very closely related to the analogous problem on the sphere. Several related problems have been studied in very different bodies of literature. On the one hand, spherical designs and spherical codes [105–

CHAPTER 4.

108] seek to place points on the sphere in order that the distance between nearest neighbors is as uniform as possible. This is related to, but not exactly the same as, the problem of packing equal sized circles on the surface of the sphere as described in [109–112]. So-called “cubature” rules [113, 114] seek to define points on the surface of the sphere so as to sample bandlimited spherical harmonic expansions and turn integrals into discrete sums (as in quadrature rules).

The issue of how to uniformly sample the rotation group, $SO(3)$, which also goes by different names in other fields (such as the “orientational space,” “Eulerian space”, etc.) has received attention in various fields from robotics and computer graphics to problems in crystallography and biomolecular structure determination. In many cases, variants of the ZXZ or ZYX Euler angles $\mathbf{q} = [\alpha, \beta, \gamma]^T$ are used in which, for example, the two Z rotations are sampled uniformly and the middle Euler angle is sampled according to a $\beta = \cos^{-1}(x)$ rule where x is sampled uniformly from $[-1, 1]$. In this way equal volume partitions of $SO(3)$ are achieved, since the integration measure for $SO(3)$ in these parameterizations has a $\sin\beta$ factor, much like the unit sphere S^2 . But equal volume partition (as studied in [115]) is not the same as equal spacing of sample points. For example, cells that are box-like regions can be long and narrow or close to cubical and still have the same volume, but the vertices of the cells in the former case would not be uniformly distributed. Therefore, placing constraints on equal volume cells is not the same as placing constraints on equal shape cells. And even having cells of equal shape does not imply that points on the vertices of these cells are uniformly spaced.

CHAPTER 4.

The Euler angles are particularly bad when it comes to uniform sampling as measured by (4.11). A variant on the Euler angles is the so-called Lattman angles [116]. These angles are defined relative to the Euler angles as $(\phi, \theta, \psi) = (\alpha + \gamma, \beta, \alpha - \gamma)$. This means that $G(\phi, \theta, \psi)$ will at least be diagonal in the computation of the cost function in (4.11), and thereby cancel with more of the identity matrix than $G(\alpha, \beta, \gamma)$, which is not even diagonal.

More recently, a number of papers have addressed how to generate close-to-uniform samplings on the sphere, torus, rotation group, and other manifolds by minimizing various energy functions [117, 118], recursive subdivisions [119, 120], interpolation using distance metrics [121, 122], quadrature-based methods [123, 124], and a fibration approach [125, 126].

These approaches all have merit, but in some cases require sophisticated recursion schemes or storing coordinates. We take a very different group-theoretic approach rather than the other approaches discussed above. Namely, we partition the rotation group into Voronoi cells, where the distance metric used is one of those reviewed in [127], and the center points of each cell is an element of one of the rotational symmetry groups of the platonic solids. Then, in the Voronoi cell centered on the identity, we construct a Cartesian grid in exponential coordinates. Because the exponential parameterization is almost linear near the identity, the distortion measure is close to zero. The contents of the Voronoi cell near the identity are then replicated in the other cells by left-shifting the contents by each rotational-symmetry operation corresponding to the symmetry group used to construct the cells. We study the shapes of these cells for the tetrahedral, octahedral, and icosahedral

CHAPTER 4.

groups. The icosahedral group is particularly interesting because it divides $SO(3)$ into smaller identical units than other groups and therefore it has the smallest distortion. This means that rotations can be sampled almost uniformly on a Cartesian grid in the Lie algebra and exponentiated.

The same concept can also be applied to the sampling on the Special Euclidean group $SE(2)$ and the subgroups of $SE(2)$, the five chiral wallpaper groups $p1$, $p2$, $p3$, $p4$, and $p6$ are used to decompose the group into Voronoi cells. In this case, $SE(2)$ is mapped to \mathbb{R}^3 with x - y axes representing translations in the x and y directions and z axis representing the rotation angle θ . The rest of the chapter is structured as follows. Sec. 4.2 reviews definition such as left-, right-, double-coset spaces in general, and the metrics used in the Lie groups $SE(2)$ and $SO(3)$. In Sec. 4.3, we talk about the details of our almost-uniform sampling method for $SO(3)$. In this section, our sampling approach of $SO(3)$ is applied to a crystal packing problem. Additionally, by introducing the sampling based on double-coset decomposition, we explain how the quality of the sample grids can be further improved. In Sec. 4.4, we provide an instructive almost-uniform sampling approach for $SE(2)$ based on coset decomposition. Finally, Sec. 4.5 is the conclusion.

4.2 Definition and Terminology

If G denotes $SO(3)$ or $SE(2)$, or any finite-dimensional Lie group, and $\Gamma, \Gamma' < G$ denote discrete subgroups, then right- and left-coset-spaces are defined as [128]

$$\Gamma \backslash G \doteq \{\Gamma g \mid g \in G\} \quad \text{and} \quad G/\Gamma' \doteq \{g\Gamma' \mid g \in G\}.$$

And a double coset space is defined as

$$\Gamma \backslash G/\Gamma' \doteq \{\Gamma g\Gamma' \mid g \in G\}.$$

Associated with any single (double-) coset, it is possible to define a set of distinguished single (double-) coset representatives, exactly one per single (double-) coset. Such a set defines a fundamental domain in G that has the same dimension as G , but lesser volume. Under the left action by Γ , the fundamental domain $F_{\Gamma \backslash G}$ is translated and the closure of the union of all translates covers G without measurable gaps² or overlaps. Similarly, right action by Γ' on the fundamental domain $F_{G/\Gamma'}$ and the double-sided-action of $\Gamma \times \Gamma'$ on $F_{\Gamma \backslash G/\Gamma'}$ produces translates the closure of which cover G .

One way to construct fundamental domains is as Voronoi cells within G . The definitions of fundamental domains $F_{\Gamma \backslash G}$, $F_{G/\Gamma'}$ and $F_{\Gamma \backslash G/\Gamma'}$ can be found in Sec. 2.1. Distance

²In practice, fundamental domains are often defined to be open sets, and so the union of translates themselves does not completely cover G , as it has gaps of measure zero. On the other hand, the union of the closure of fundamental domains will cover, but with a set of measure zero of duplicates. The distinction between a fundamental domain, its interior, and its closure are inconsequential for our purposes.

CHAPTER 4.

functions for $SO(3)$ and $SE(2)$ can be define as

$$d_{SO(3)}^{(1)}(R_1, R_2) = \|R_1 - R_2\| \quad (4.2)$$

where $\|A\| = \sqrt{\text{tr}(AA^T)}$ is the Frobenius norm, and

$$d_{SE(2)}^{(1)}(g_1, g_2) = \|g_1 - g_2\|_W \quad (4.3)$$

where an arbitrary element of $SE(2)$ is of the form

$$g = \begin{pmatrix} \cos \theta & -\sin \theta & x \\ \sin \theta & \cos \theta & y \\ 0 & 0 & 1 \end{pmatrix},$$

$W = W^T$ as a 3×3 positive definit weighting matrix, and $\|A\|_W = \sqrt{\text{tr}(AWA^T)}$. The above distance measures are ‘extrinsic’ in the sense that they rely on how these matrix Lie groups are embedded in $\mathbb{R}^{3 \times 3}$, but they satisfy the conditions of non-negativeness, symmetry, and the triangle inequality.

It is also possible to defin ‘intrinsic’ measures of distance using the logarithm function. Since both $SO(3)$ and $SE(2)$ are matrix-Lie-groups, their exponential maps are the matrix exponentials, and the corresponding inverse map for each is the matrix logarithm (see the details in Sec. 2.1). This degenerates when $\|\mathbf{x}\|$ or θ is π . By restricting the discussion to the case when $\|\mathbf{x}\|, \theta < \pi$, \log is uniquely define on a subset of $SO(3)$ depleted by a set of measure zero. This depletion will have no effect on our formulation. For example, it becomes possible to defin

$$d_{SO(3)}^{(2)}(R_1, R_2) = \|\log(R_1^T R_2)\|$$

CHAPTER 4.

when $R_1^T R_2$ is not a rotation by π , and otherwise $d_{SO(3)}^{(2)}(R_1, R_2) = \pi$ and similarly

$$d_{SE(2)}^{(2)}(g_1, g_2) = \|\log(g_1^{-1} \circ g_2)\|_{W'}$$

where W' could be different than W . As in the $SO(3)$ case, a map from $se(2)$ to \mathbb{R}^3 can be define as $X^\vee = [v_1, v_2, \theta]^T$.

4.3 An Almost-Uniform Sampling Approach for $SO(3)$

4.3.0.3 Discrete subgroups of $SO(3)$

There are two families of discrete subgroups of the rotation group $SO(3)$. One family of discrete subgroups corresponds to rotational symmetry operations of the Platonic solids—the tetrahedral group T , the octahedral group O , and the icosahedral group I . Another family of discrete subgroups of $SO(3)$ are the cyclic groups C_n which result from sampling rotations around a fixed axis at angles of $2\pi/n$. The rotational symmetry groups of the platonic solids can be described in terms of cyclic groups of different orders applied to their faces, edges, and vertices so as to leave the volume occupied by the solids unchanged. In our sampling method, we only be concerned with the groups of rotational symmetries of the Platonic solids since they fill $SO(3)$ more uniformly than other subgroups.

Here we review some details of the rotational symmetries of the Platonic solids. A regular tetrahedron has four C_3 axes through each vertex to the center of the opposite

CHAPTER 4.

face, and 3 C_2 axes through the centers of pairs of opposite edges. Including the identity, the total number of group elements of T is $4(3 - 1) + 3(2 - 1) + 1 = 12$. A cube and its dual, the octahedron, have 3 C_4 axes through the centers of opposite faces, 4 C_3 axes through opposite vertices, and 6 C_2 axes through the centers of pairs of opposite edges. The total number of group elements of O is $3(4 - 1) + 4(3 - 1) + 6(2 - 1) + 1 = 24$. A regular icosahedron and its dual dodecahedron have 6 C_5 axes through opposite vertices, 10 C_3 through the centers of opposite faces, and 15 C_2 axes through the centers of opposite edges. Thus, the subgroup I has $6(5 - 1) + 10(3 - 1) + 15(2 - 1) + 1 = 60$ elements.

4.3.0.4 Generating Voronoi cells of $SO(3)$

In this chapter, we parameterize $SO(3)$ as a solid ball in \mathbb{R}^3 with radius π in exponential coordinates (see the details in Sec. 2.1). $SO(3)$ is decomposed into Voronoi cells centered on elements of its discrete subgroup Γ (Γ can be T , O or I), with each Voronoi cell consisting of the points contained in $SO(3)$ closer to one element of Γ than to any other. A common metric to define the distance between any two points $R_1, R_2 \in SO(3)$ is

$$d(R_1, R_2) = \theta(R_1^T R_2) = \|(\log(R_1^T R_2))^\vee\| \quad (4.4)$$

where $\log(\cdot)$ is the inverse of the exponential mapping in (2.24), which is valid when $\theta \in [0, \pi)$. This is extended to the closed solid ball by simply setting $\theta(R_1^T R_2) = \pi$ on the boundary. Here, as earlier, $\|\cdot\|$ is the Euclidean norm, and \vee is defined by the property that $(\hat{\mathbf{x}})^\vee = \mathbf{x}$.

CHAPTER 4.

The Voronoi cells of $SO(3)$ represented in the exponential coordinates can partition the solid ball in \mathbb{R}^3 into separate regions, denoted as S_1, S_2, \dots, S_m , where m is the number of elements in Γ (see Fig. 4.1 for the octahedral group). These regions are mildly ambiguous at their borders, but this represents a set of measure zero. The region S_1 located in the center of the solid ball corresponds to the Voronoi cell of the identity matrix. Because the exponential parameterization is relatively flat near the identity, i.e., $\exp \hat{\mathbf{x}} \approx \mathbb{I} + \hat{\mathbf{x}}$ when $\|\mathbf{x}\| \ll 1$, the metric tensor becomes $G(\mathbf{x}) \approx \mathbb{I}$, and the distortion measure $C(\mathbf{x})$ of the samples for $SO(3)$ parametrized by Cartesian grids on the S_1 is close to zero. And the smaller size of S_1 , the smaller the overall distortion will be. Subsequently, the samples of other Voronoi cells of $SO(3)$ can be obtained from the samples computed for the “center” Voronoi cell (i.e., the one that corresponds to the identity element of $SO(3)$ and Γ) by multiplying with the other rotation matrices in Γ . In other words, after we generate N samples for the center Voronoi cell, for every other Voronoi cell $S_k \subset SO(3)$, we replicate the N samples by shifting those from the center cell, S_1 . This can be done by either left or right shift:

$$\{\exp(\hat{\mathbf{x}}_{k_i}) | i = 1, \dots, N\} = \{\exp(\hat{\mathbf{x}}_{1_i}) R_k | i = 1, \dots, N\} \quad (4.5)$$

or

$$\{\exp(\hat{\mathbf{x}}'_{k_i}) | i = 1, \dots, N\} = \{R_k \exp(\hat{\mathbf{x}}_{1_i}) | i = 1, \dots, N\}, \quad (4.6)$$

for all $k = 2, \dots, m$ where $\{\hat{\mathbf{x}}_{k_i} | i = 1, \dots, N\}$ and $\{\hat{\mathbf{x}}'_{k_i} | i = 1, \dots, N\}$ are the N samples for the center Voronoi cell generated by these shifts, and R_k is the element of the discrete subgroup Γ . Note that while in general at the level of individual samples $\mathbf{x}_{k_i} \neq \mathbf{x}'_{k_i}$, at the

CHAPTER 4.

level of whole cells

$$\overline{S}_k = R_k \cdot \overline{S}_1 = \overline{S}_1 \cdot R_k$$

where \overline{S}_k denotes the closure of cell S_k and \cdot denotes the application of R_k to each point in \overline{S}_k .

Thus, without loss of generality, we can focus on sampling the center Voronoi cell, and samples of the rest of $SO(3)$ can be obtained by shifting the samples from the center one, without introducing more distortion. The reason why we can shift from either the left or the right is that the metric in (4.4) used to generate the Voronoi cells is left and right invariant, i.e., $d(R_1, R_2) = d(AR_1, AR_2) = d(R_1A, R_2A)$ for any $A \in SO(3)$. Therefore, the Voronoi cells that we are considering can be viewed as the “fundamental domains” constructed from representative group elements that in total correspond to the left and right coset spaces $SO(3)/\Gamma$ and $\Gamma \backslash SO(3)$.

In this chapter, the shape of Voronoi cells for the tetrahedral group, the octahedral group, and the icosahedral group are studied. The discrete subgroup with the smallest Voronoi cells should be used for the sampling to minimize the distortion; a small-sized cell centered on the identity will be good for almost-uniform sampling because near the identity, $\exp \hat{\mathbf{x}} \approx \mathbb{I} + \hat{\mathbf{x}}$, and with \mathbf{x} sampled uniformly in Cartesian coordinates, exponentiation does not warp the resulting rotations very much. Since the icosahedral group I has the largest number of group elements, $SO(3)$ is divided into smallest Voronoi cells compared to other discrete subgroups. Thus, to achieve best sampling results, the icosahedral group is used to generate the Voronoi cells in $SO(3)$. Fig. 4.2 shows that the center Voronoi cell

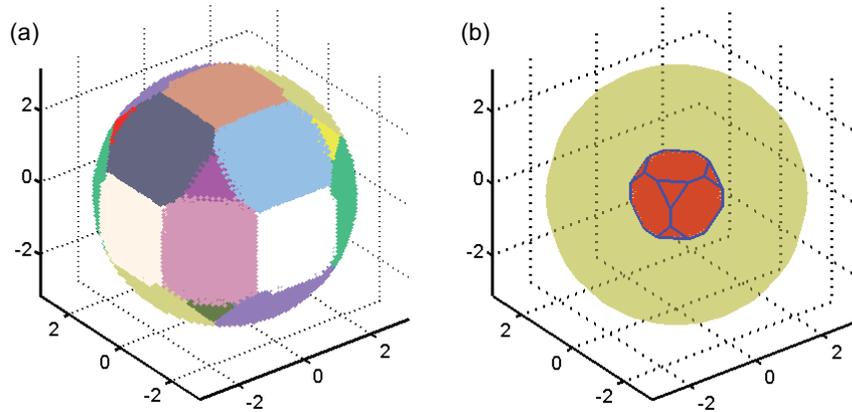


Figure 4.1: (a) Voronoi cells of the octahedral group in $SO(3)$ are represented in exponential coordinates. Each color represents a Voronoi cell in $SO(3)$. We note that because the exterior Voronoi cells cover the interior ones in the figure the regions with different colors may appear to be patches on a sphere but actually represent for sites in a solid ball. (b) The center Voronoi cell that corresponds to the identity matrix is represented in exponential coordinates. The yellow-shaded ball represents the solid ball in \mathbb{R}^3 with radius π .

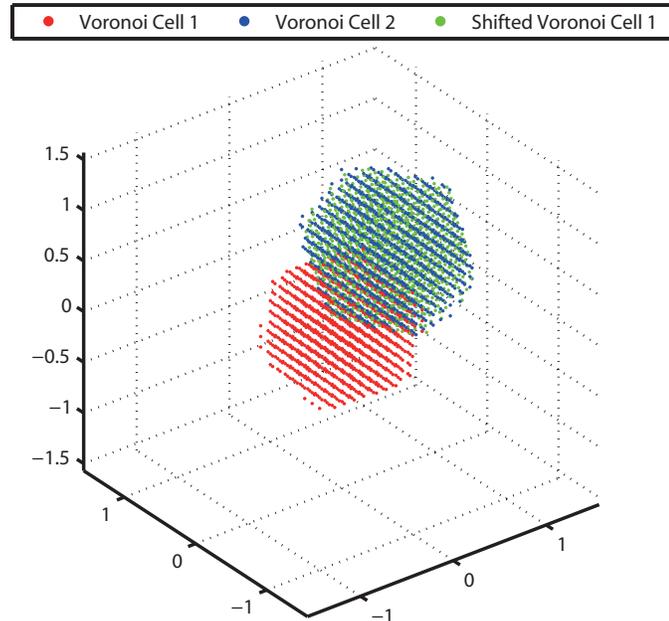


Figure 4.2: The center Voronoi cell shifted by a rotation matrix R in the icosahedral group VS the Voronoi cell that corresponds to R . The complete overlapping between the two sites shows that the contents of the Voronoi cell on R can be replicated by the contents of the Voronoi cell on the identity by shifting the contents by R .

CHAPTER 4.

shifted by a rotation matrix in the icosahedral group is overlapped with the Voronoi cell that corresponds to this rotation matrix. The Voronoi cells are visualized in the exponential coordinates.

The shape of the Voronoi cells centered on the identity for the tetrahedral, octahedral and icosahedral groups are close to being polyhedra themselves; more specifically, they are almost cropped octahedron, cropped cube and dodecahedron [129]. We say “almost polyhedra” because the boundaries between the central cell and its neighbors are slightly curved inward, making the cell slightly rounded. But by considering the polyhedra that enclose the actual Voronoi cells, we can be guaranteed to cover $SO(3)$ with samples without gaps. And redundant points can easily be removed.

For the tetrahedral group, the center Voronoi cell is an octahedron cropped by surfaces normal to the axes through opposite vertices, with the distance $h_{C_2} = \pi/2$ from the origin (see Fig. 4.3 (a1)). The Cartesian coordinates of the vertices on the cropped octahedron are

$$\begin{aligned}
 & (\pm\sqrt{3}h_{C_3} \mp (\sqrt{3}h_{C_3} - h_{C_2}), \pm(\sqrt{3}h_{C_3} - h_{C_2}), 0), & (4.7) \\
 & (\pm\sqrt{3}h_{C_3} \mp (\sqrt{3}h_{C_3} - h_{C_2}), 0, \pm(\sqrt{3}h_{C_3} - h_{C_2})), \\
 & (\pm(\sqrt{3}h_{C_3} - h_{C_2}), \pm\sqrt{3}h_{C_3} \mp (\sqrt{3}h_{C_3} - h_{C_2}), 0), \\
 & (0, \pm\sqrt{3}h_{C_3} \mp (\sqrt{3}h_{C_3} - h_{C_2}), \pm(\sqrt{3}h_{C_3} - h_{C_2})), \\
 & (\pm(\sqrt{3}h_{C_3} - h_{C_2}), 0, \pm\sqrt{3}h_{C_3} \mp (\sqrt{3}h_{C_3} - h_{C_2})), \\
 & (0, \pm(\sqrt{3}h_{C_3} - h_{C_2}), \pm\sqrt{3}h_{C_3} \mp (\sqrt{3}h_{C_3} - h_{C_2})).
 \end{aligned}$$

CHAPTER 4.

For the octahedral group, the center Voronoi cell is a cube with the center-to-face distance $h_{C_4} = \pi/4$, cropped at each corner by a surface of a distance $h_{C_3} = \pi/3$ from the origin (see Fig. 4.3 (b1)). The Cartesian coordinates of the vertices on the cropped cube are

$$h_{C_4}(\pm 1 \mp t, \pm 1, \pm 1) \quad (4.8)$$

$$h_{C_4}(\pm 1, \pm 1, \pm 1 \mp t)$$

$$h_{C_4}(\pm 1, \pm 1 \mp t, \pm 1),$$

where $t = 3\pi/4 - \sqrt{3}h_{C_3}$.

For the icosahedral group, the center Voronoi cell is a dodecahedron with the center-to-face distance $h_{C_5} = \pi/5$ (see Fig. 4.3 (c1)). The Cartesian coordinates of the vertices on the dodecahedron are

$$c(\pm 1, \pm 1, \pm 1), \quad (4.9)$$

$$c(0, \pm 1/\varphi, \pm \varphi),$$

$$c(\pm 1/\varphi, \pm \varphi, 0),$$

$$c(\pm \varphi, 0, \pm 1/\varphi),$$

where $c = 20h_{C_5}/(\sqrt{250 + 110\sqrt{5}}(\sqrt{5} - 1))$.

4.3.0.5 Curvature of the boundaries of the Voronoi cells for $SO(3)$

Here we consider the shape of the boundary between the cell S_0 and adjacent cell S_k centered on $\gamma_k = \exp(\theta_k \hat{\mathbf{n}}_k)$. This boundary is defined by the set of all $\mathbf{x} \in \mathbb{R}^3$ defined by

CHAPTER 4.

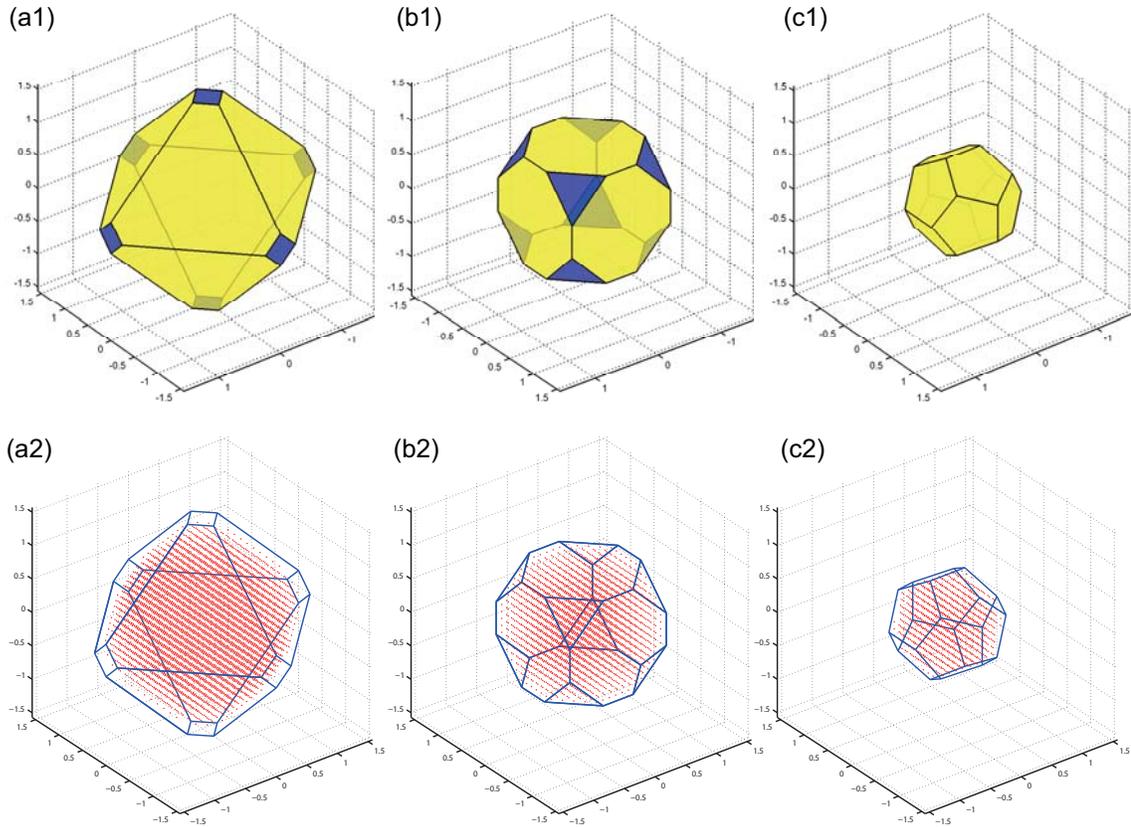


Figure 4.3: Polyhedra in $\mathbb{R}^3 \cong so(3)$ depicting the center Voronoi cells in $SO(3)$ corresponding to the (a1) tetrahedral (b1) octahedral and (c1) icosahedral groups. The wire frame plots of these polyhedra with an example of uniform grids in \mathbb{R}^3 filled in for the (a2) tetrahedral, (b2) octahedral and (c2) icosahedral groups.

CHAPTER 4.

the condition

$$d(\mathbb{I}, \exp(\hat{\mathbf{x}})) = d(\exp(\hat{\mathbf{x}}), \gamma_k). \quad (4.10)$$

When $\theta_k < 1$, this condition can be approximated by $\|\mathbf{x}\| = \|\mathbf{x} - \theta_k \mathbf{n}_k\|$, which define a plane passing through $(\theta_k/2)\mathbf{n}_k$ with normal in the direction of \mathbf{n}_k . However, when θ_k is not small, (4.10) describes a surface with potentially significant curvature. This surface curves inward toward the origin from the plane described above, with which it shares the single point $\mathbf{x} = (\theta_k/2)\mathbf{n}_k$. This surface is a surface of revolution with axis \mathbf{n}_k . This can be seen from (4.10) and the bi-invariance of the metric, which implies invariance under conjugation (similarity transformations) as well. In particular, applying a similarity transformation to each entry in (4.10) with respect to $Q(\phi) = \exp(\phi \hat{\mathbf{n}}_k)$ where ϕ is an arbitrary angle of rotation gives

$$d(\mathbb{I}, \exp(\widehat{Q(\phi)\mathbf{x}})) = d(\exp(\widehat{Q(\phi)\mathbf{x}}), \gamma_k)$$

because $(Q(\phi)\hat{\mathbf{x}}Q^T(\phi))^\vee = Q(\phi)\mathbf{x}$ and $Q(\phi)\gamma_kQ^T(\phi) = \gamma_k$. Since these boundaries curve in, polyhedral descriptions of the cells are conservative, in that they contain all sample points in S_k , and can be depleted appropriately so as to avoid redundancy in sampling.

4.3.0.6 Distortion Measure of the samplings for $SO(3)$

In this section, we calculate the distortion of our proposed sampling method and compare it with a widely used approach, the sampling based on Euler angles parametrization.

CHAPTER 4.

Both methods are evaluated as (4.11),

$$C(\mathbf{q}) = \frac{1}{\sqrt{3}} \|G(\mathbf{q}) - \mathbb{I}\| \quad (4.11)$$

and $G(\mathbf{x}) = J_r^T(\mathbf{x})J_r(\mathbf{x}) = J_l^T(\mathbf{x})J_l(\mathbf{x})$, where the Jacobians for $SO(3)$ in exponential parameterization $R(\mathbf{x}) = \exp(\hat{\mathbf{x}})$ can be found in Sec. 2.2.

In our sampling approach, the distortion by using the icosahedral group is shown in Fig. 4.4 (a). We can see that since the exponential parameterization is almost flat near the identity, when $\|\mathbf{x}\| \leq 0.1$, the distortion is almost zero. As $\|\mathbf{x}\|$ increases, the distortion becomes larger. But even at the furthest point of the center Voronoi cell, $\|\mathbf{x}\| = 0.73$, the distortion is only 3.52%. Since shifting the center Voronoi cell by rotational symmetry operation introduces no more distortion, it is the largest sampling distortion for $SO(3)$ and independent of the resolution of the Cartesian grids.

Using the same measure, we also evaluate the distortion for ZYZ Euler angles (see Fig. 4.4 (b)). The corresponding Jacobians can also be found in Sec. 2.2. The two Z rotations α and γ are sampled uniformly and the middle Euler angle β is sampled based on $\beta = \cos^{-1}(t)$, where t is sampled uniformly. The distortion remains the same using either the left or the right Jacobian. By using the sampling method based on Euler angles, even if the equal volume partitions of $SO(3)$ are achieved, we can see that the distortion is significantly large. As the value of β close to the singularity, i.e., $\beta = \pi$, the distortion increases and reaches the peak value 81.65%.

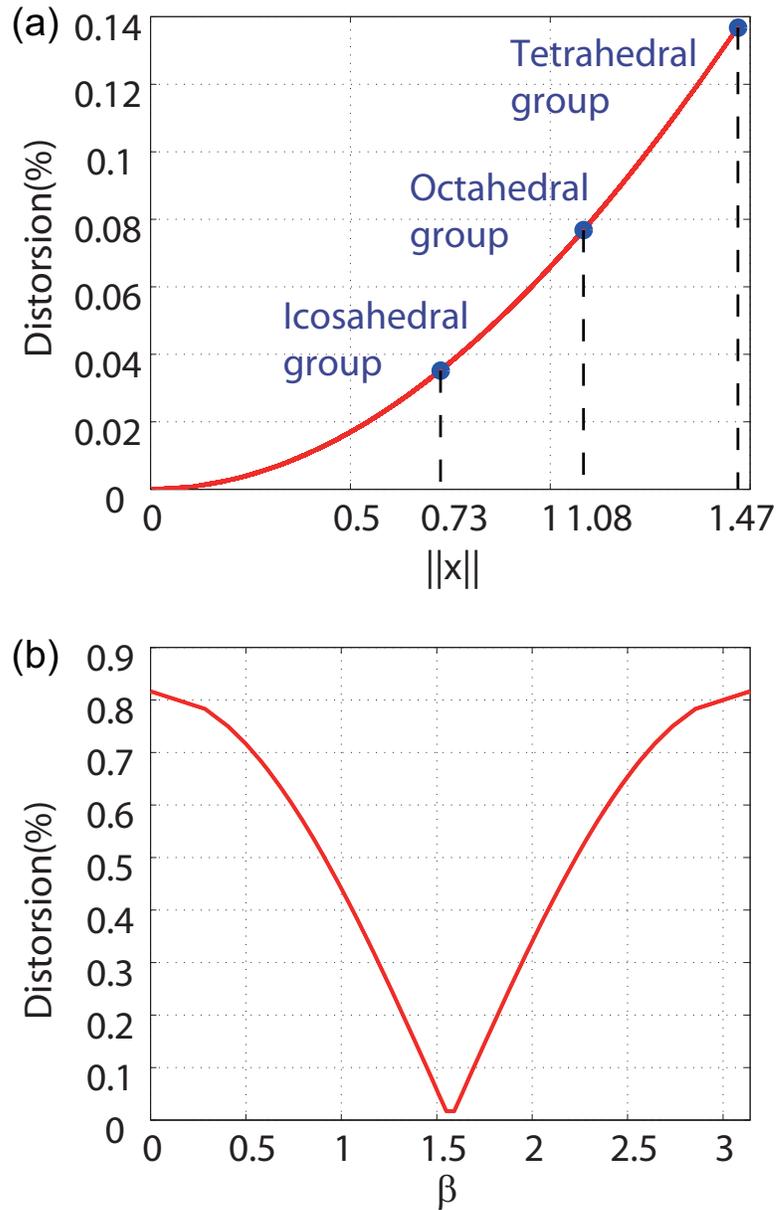


Figure 4.4: The distortion of the samples on $SO(3)$ using (a) our sampling method and (b) ZXZ Euler angles. In our sampling method, all the three discrete subgroups are evaluated on the norm of the Cartesian coordinates under exponential parameterization, $\|x\|$. The maximum distortion for each discrete subgroup is marked in the figure. The icosahedron group should be used since it has the smallest Voronoi cell and results in the smallest distortion. By using the ZXZ Euler angles, the two Z rotations α and γ are sampled uniformly and the middle Euler angle β is sampled based on $\beta = \cos^{-1}(t)$, where t is sampled uniformly. In this case, the distortion is evaluated on each value of β from 0 to π .

4.3.1 Application in the context of a crystal packing problem

In many fields such as attitude estimation, medical image registration, and robot motion planning, generating a good set of samples is also very important, particularly in the operations such as optimization, conformation searching, and path generation. For most stochastic optimization methods, the quality of sampling has significant impact on their performance. In the conformation search and path generation, an efficient sampling method can substantially increase the computational speed [126].

In this section, we demonstrate our sampling method in the problem of computing collision-free packing arrangements in a crystal. This is important because this problem is closely related to X-ray crystallography. X-ray crystallography is one of the most used methods for determining protein structures and associated complexes. In order to interpret the information contained in an X-ray diffraction pattern when the shape of fragments of the proteins in crystal are known in advance, it is critical to find unknown rigid-body motions that relate the fragments to each other. The computational problem of finding these rigid-body motions is known as ‘molecular replacement’ and algorithms for solving the problem have been investigated for half a century [130, 131]. Molecular replacement seeks to place a homologous/similar molecule in the crystallographic unit cell and search in the whole configuration space so as to maximize the correlation with x-ray diffraction data (see Chap. 3 for a complete description of the MR method). Obviously, when performing a search over

CHAPTER 4.

rotations, one desires the samples to be generated as ‘uniformly’ as possible, since having samples clumped in some areas and sparse in others will be a waste of computational resources.

In this crystal packing problem, both collision checking and stochastic searching are involved, and their performance heavily relies on the sampling quality in the conformation space. To evaluate our sampling method, we look into a simplified toy model that simulates the packing of protein crystals (similar to the one used in Chap. 3). The rabbit-shaped toy model has fixed position and can rotate freely in space; in other words, the motion of the toy model is in $SO(3)$. The copies of the model tile the space in P1 symmetry, in which the copies retain the same motion (translation and rotation) with their positions shifted by the unit cell (see Fig. 4.5). Our goal is to obtain a sufficient candidate set of collision-free packing arrangements by using as few conformation searches as possible. We generate a similar number of samples on $SO(3)$ using both our proposed method and ZXZ Euler angles. In Fig. 4.6, we can see that when the number of samples is small (≤ 500), using Euler angles, no collision-free conformation is found while 42 are found using our sampling method. Even as the number of samples increases, we can always find more collision-free conformations using our method than sampling by Euler angles. The efficiency in finding collision-free candidates using our sampling method comes from the fact that the samples we generate in $SO(3)$ are more evenly spaced and have less distortion. Besides this protein crystal packing problem, our sampling method also has great potential in robot motion planning problem, which involves finding feasible conformations in high

dimensional space.

4.3.2 Improved almost-uniform sampling on $SO(3)$ based on double-coset decompositions

From previous sections, we know that by using single-coset decompositions, we can decompose the $SO(3)$ into 60 identical small units and the smaller the units are, the less metric distortion we can achieve. In this section, we talk about how to improve the quality of the samples on $SO(3)$ by decomposing $SO(3)$ into even smaller units based on double-coset decompositions.

Given two finite subgroups $H, K < G$ where $G = SO(3)$, and the condition $|H \cap K| = 1$, the nonoverlapping tiles generated by the action of $H \times K$ on $F_{H \backslash G / K}$, the fundamental domain of the double-coset space $H \backslash G / K$ (defined in Sec. 2.1), satisfy

$$G = \bigcup_{h \in H} \bigcup_{k \in K} h \overline{F_{H \backslash G / K}} k^{-1}. \quad (4.12)$$

Some examples of double-coset spaces are given in Fig. 4.7 with K taken as the icosahedral group for all cases and H taken as the conjugated tetrahedral group (a), the conjugated octahedral group (b) and the conjugated icosahedral group (c), respectively, where the conjugated group H with respect to the original group H_0 is defined as $H = gH_0g^{-1}$ for $g \in G$. In all cases conjugation is taken with respect to an element of G that is not in H_0 or K . In all of these figures the shaded regions are the fundamental domains for the double-coset spaces, the yellow dodecahedron is the fundamental domain for the single-

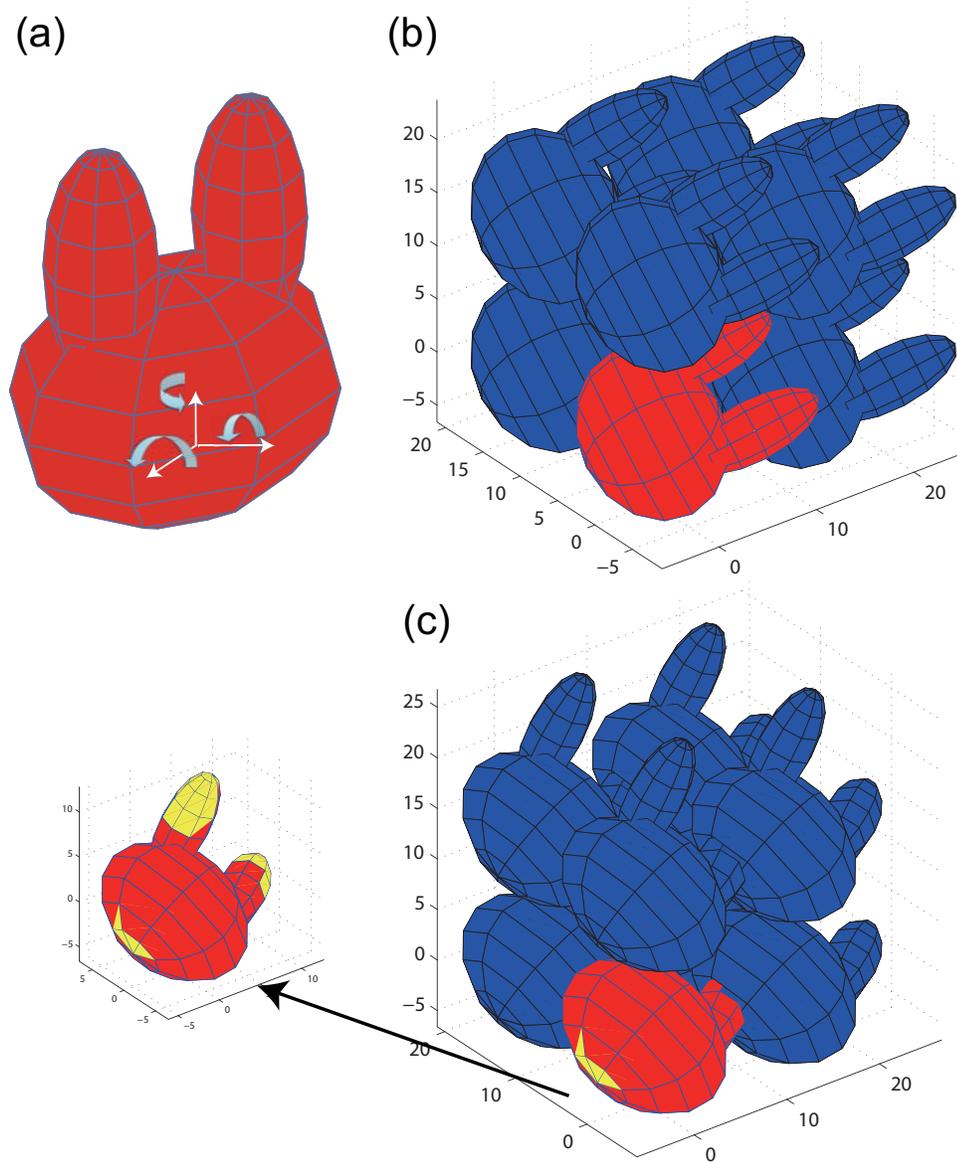


Figure 4.5: (a) Illustration of 3 degrees of freedom in the packing model. Examples of packing configuration (b) without collisions (c) with collisions (indicated in yellow) (the same model is used in Chap. 3).

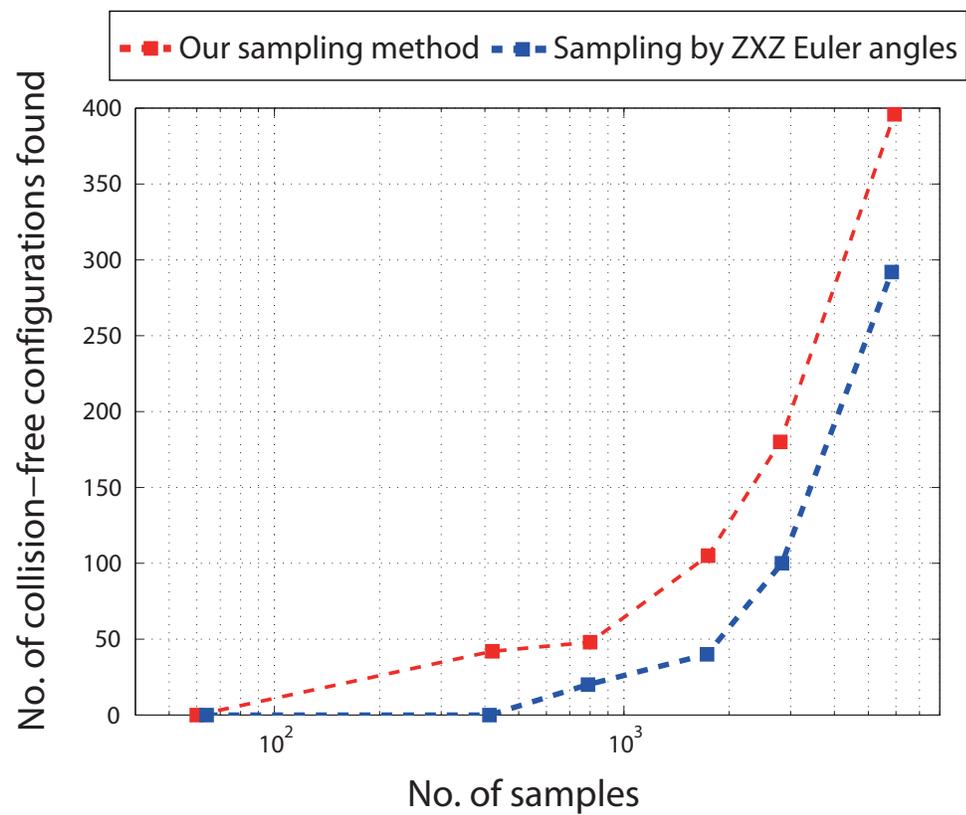


Figure 4.6: The number of collision-free conformations VS the number of sample points on $SO(3)$ generated by (a) our sampling method, (b) sampling of Euler angles.

CHAPTER 4.

coset space of $SO(3)$ based on the icosahedral group (see Fig. 4.3 (c1)). The figure is illustrated in exponential coordinates, with $SO(3)$ represented as a solid ball of radius π . Fig. 4.8 demonstrates that when H is an icosahedral group, K is a conjugated icosahedral group, i.e., $K = g \circ H \circ g^{-1}$, where $g \in SO(3)$, and $H \cap K = \{e\}$, these $|H| \cdot |K| = 60 \times 60 = 3600$ Voronoi cells can tile G , i.e., $SO(3)$ without gaps.

A great advantage to use the double-coset space to sample $SO(3)$ is that it can result in less metric distortion. Since the exponential parametrization is almost linear near the identity, i.e., $\exp \hat{\mathbf{x}} \approx \mathbb{I} + \hat{\mathbf{x}}$ when $\|\mathbf{x}\| \ll 1$, the metric tensor becomes $J^T(\mathbf{x})J(\mathbf{x}) \approx \mathbb{I}$, and the distortion measure $C(\mathbf{x})$ of the samples on $SO(3)$ parametrized by Cartesian grids on the center Voronoi cell is close to zero. Therefore, as $|H| \cdot |K|$ increases and the size of the center Voronoi cells shrinks, the smaller the overall distortion will be. In Fig. 4.9, we can see the maximum distortion is only 0.0074 using the double-coset space with H as the icosahedral group and K as a conjugated icosahedral group, which is only 21% of the smallest maximum distortion from the single-coset space, i.e., when one copy of the icosahedral group is used.

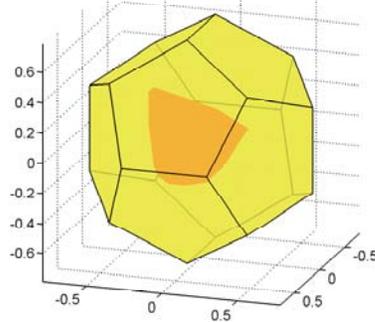
When $H = K$, it is possible to construct $F_{H \setminus G/H}$, but not in the way described above. In this scenario, the way to interpret $F_{H \setminus G/H}$ is as a barycentric subdivision of $F_{H \setminus G}$ such that the original coset space can be reconstructed (up to a missing set of measure zero) by the following combination of adjoint action and union

$$\overline{F_{H \setminus G}} = \bigcup_{h \in H} h \overline{F_{H \setminus G/H}} h^{-1}$$

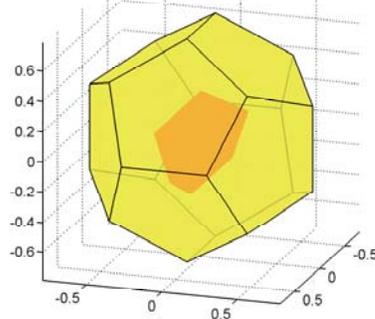
For example, if H is the group of rotational symmetry operations of the icosahedron, and

CHAPTER 4.

(a) H =tetrahedral group (conjugated), K = icosahedral group



(b) H =octahedral group (conjugated), K = icosahedral group



(c) H =icosahedral group (conjugated), K = icosahedral group

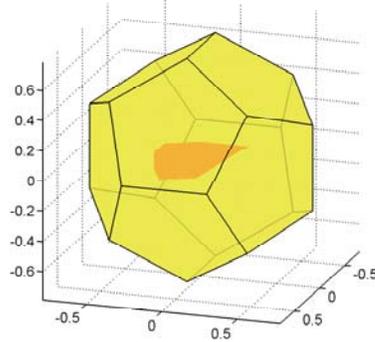


Figure 4.7: The center Voronoi cell in a single-coset-space decomposition ($F_{K \setminus SO(3)} = F_{SO(3)}/K =$ yellow-shaded region) with K as the icosahedral group, and the center Voronoi cells in double-coset-space decompositions (red-shaded regions) with K as the icosahedral group for all cases and H as the conjugated tetrahedral group (a), the conjugated octahedral group (b) and the conjugated icosahedral group (c), respectively.

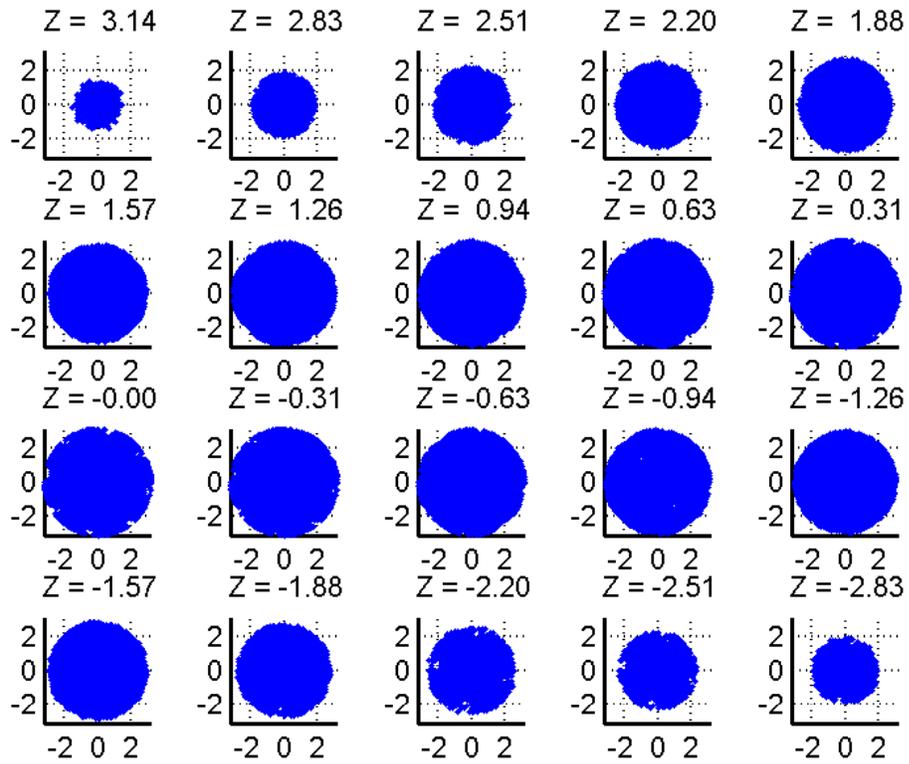


Figure 4.8: The sliced plots of the union of the Voronoi cells from the double-coset space $H\backslash G/K$, where H is the icosahedral group, K is a conjugated version of the icosahedral group and G is $SO(3)$. Here $K = g \circ H \circ g^{-1}$, where $g \in SO(3)$, and $H \cap K = \{e\}$.

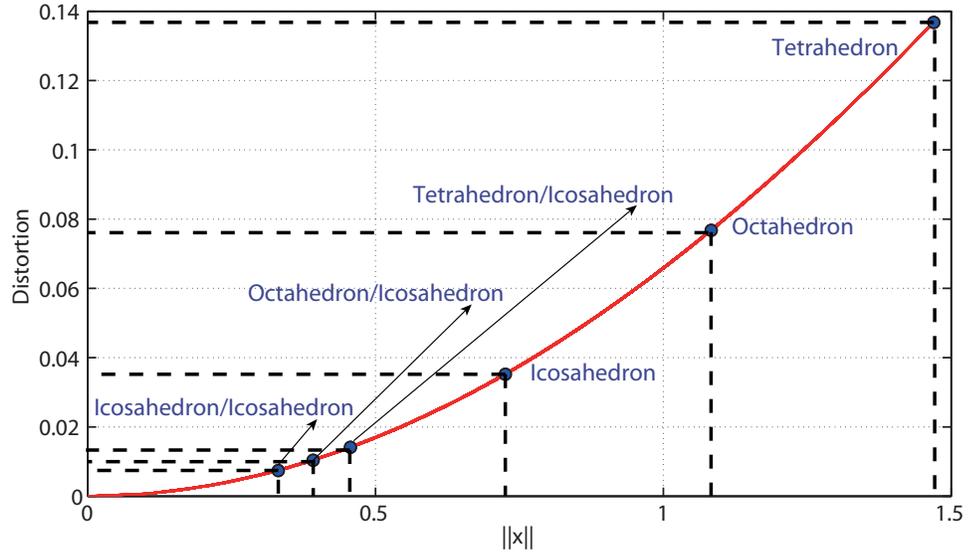


Figure 4.9: The maximum distortions from different double and single coset spaces.

$G = SO(3)$, then $|H| = 60$ and $\log(F_{H \setminus G})$ can be viewed as a dodecahedral cell (see Fig. 4.10) centered at the origin of the Lie algebra, $\mathcal{G} = so(3)$, and each $F_{H \setminus G/H}$ can be viewed as an irregular tetrahedron (the red-shaded region in Fig. 4.10).

Similarly, if $K < H$, then a $|K|$ -fold division of $F_{H \setminus G}$ can be computed to represent $F_{H \setminus G/K}$ and these pieces can be reconstructed by adjoint action, which has the effect of rotation in $so(3)$ since $(RXR^T)^\vee = Rx$.

This means that when using two copies of the icosahedral group, we can divide $SO(3)$ into 3600 tetrahedral pieces of equal size and shape. A reasonably uniform and fine sampling can be obtained by choosing the point at the center of each of these tetrahedra (and at their vertices if desired). If even finer samples are required, then sampling from an arbitrarily fine Cartesian grid in the Lie algebra can be imposed within each tetrahedron.

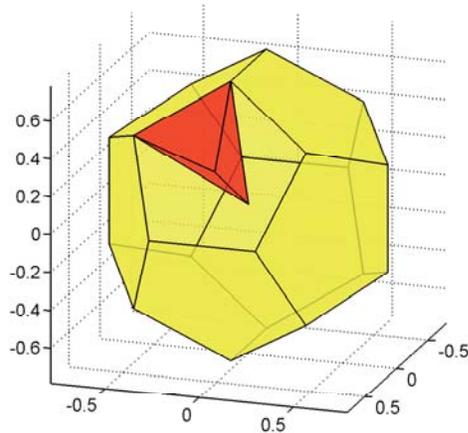


Figure 4.10: The dodecahedral cell (yellow-shaded region) and the tetrahedral wedge (red-shaded region). The dodecahedral cell can be decomposed into 60 identical tetrahedral wedges like this, with 5 packed to form a pyramid with pentagonal base corresponding to a face of the dodecahedron.

4.4 An Instructive Approach for Almost-Uniform Sampling for $SE(2)$

In this section, we establish for the first time what the fundamental domains look like when $G = SE(2)$ and Γ is one of the five chiral wallpaper groups. $SE(2)$ is mapped to \mathbb{R}^3 with x - y axes representing translations in the x and y directions and z axis representing the rotation angle θ . The Voronoi cells are generated using the metric (4.3). Fig. 4.11 illustrates the lattice structure for wallpaper group $p1$ and the corresponding Voronoi cells of $SE(2)$. The Voronoi cells centered at the identity for all the five wallpaper groups $p1$, $p2$, $p4$, $p3$ and $p6$ are shown in Fig. 4.12. The group $p1$ consists only of translations, in a parallelogrammatic lattice. Its center Voronoi cell looks like a hexagonal box with the height from $-\pi$ to π . We note that when the lattice is square, the center Voronoi cell

CHAPTER 4.

becomes a square box. The group $p2$ differs only from $p1$ in that it contains 180° rotations, or rotations of order 2, so its center Voronoi cell has the same hexagonal shape in $x - y$ cross section, but the height is from $-\pi/2$ to $\pi/2$, reduced by half along the θ -axis. $p4$ is the group with a 90° rotation, in a square lattice, so it has square-shaped center Voronoi cell, with the height from $-\pi/4$ to $\pi/4$, further cut by half from $p2$. $p3$ and $p6$ are the symmetry groups for a hexagonal lattice, with a 120° rotation and a 60° rotation, respectively. So they have regular hexagonal-shaped center Voronoi cells, with the height from $-\pi/3$ to $\pi/3$ and from $-\pi/6$ to $\pi/6$, respectively.

This discussion of the 2D case is instructive. As can be seen, the cells can be viewed as having the same cross section for different values of θ . The spatial generalization of this is that if $G = \mathbb{R}^3 \rtimes SO(3)$ and $\Gamma = \mathbb{Z}^3 \rtimes \mathbb{P}$ where $\mathbb{P} < SO(3)$ is the crystallographic point group and $\mathbb{Z}^3 < \mathbb{R}^3$ is the lattice translation group, then

$$F_{\Gamma \backslash G} \cong F_{\mathbb{Z}^3 \backslash \mathbb{R}^3} \times F_{\mathbb{P} \backslash SO(3)}.$$

It has been known for more than a century that of the 230 space groups, 65 are chiral ones, and only these occur in protein crystallography. Of these, 24 can be written as semi-direct products, as above. These are called symmmorphic, and the other 41 are called nonsymmorphic. In the planar case discussed above, all five chiral wallpaper groups are symmmorphic. Though motivated by the symmmorphic case, in general $F_{\Gamma \backslash G}$ can be decomposed into a product of something akin to $F_{\mathbb{P} \backslash SO(3)}$ with a sample space of translations (though in the nonsymmorphic case it will not be as simple as $F_{\mathbb{Z}^3 \backslash \mathbb{R}^3}$). For this reason we investigate almost-uniform sampling on $F_{\mathbb{P} \backslash SO(3)}$ by further subdividing it using double-coset decom-

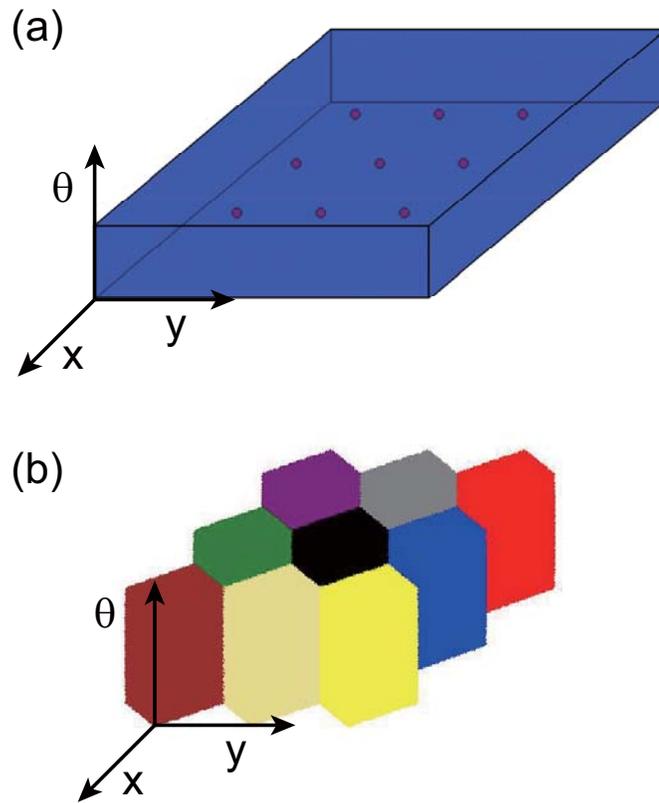


Figure 4.11: (a) $SE(2)$ illustrated in \mathbb{R}^3 and the parallelogrammatic lattice for $p1$; (b) the Voronoi cells for $SE(2)$ based on $p1$.

CHAPTER 4.

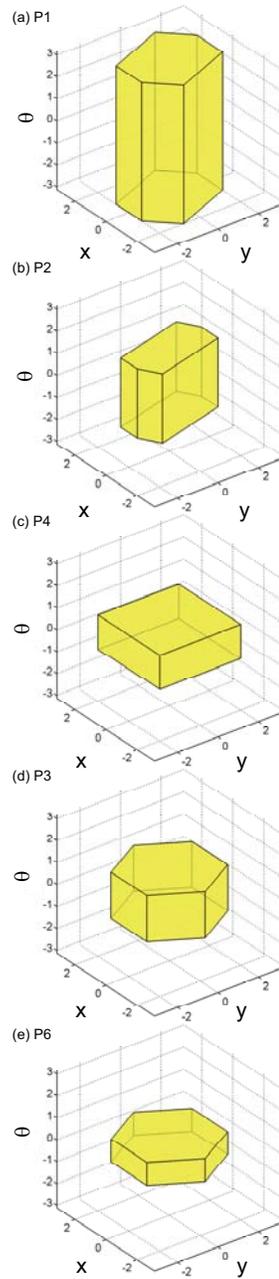


Figure 4.12: $F_{\Gamma_i \setminus SE(2)}$ for 5 chiral wallpaper groups $p1$ (a), $p2$ (b), $p4$ (c), $p3$ (d) and $p6$ (e).

positions.

4.5 Conclusion

A new sampling method on the rotation group $SO(3)$ is proposed. In this method, $SO(3)$ is partitioned into Voronoi cells based on the rotational symmetry operations of the Platonic solids. Uniform Cartesian grids in exponential coordinates are generated on the Voronoi cell centered on the identity and samples on the surrounding Voronoi cells are then replicated by shifting by the rotational symmetry operations. The shape of the Voronoi cells centered on the identity for the tetrahedral, octahedral and icosahedral groups are close to being polyhedra themselves. For the icosahedral group, the one has the smallest Voronoi cells, the largest distortion of the samples is only 3.52, independent of resolution of the Cartesian grid. Moreover, that distortion is further reduced to 0.0074 using the doublecoset space decomposition based on the icosahedral group and a conjugated icosahedral group. This sampling method can achieve almost uniform sampling at any level of resolution, without having to store large numbers of coordinates or requiring sophisticated data structures. In the implementation to a crystal packing problem, the sampling method shows improved efficiency in finding collision-free conformations compared to sampling by Euler angles and it can also apply to robot motion planning problem with great potential. Additionally to the sampling on $SO(3)$, we also provide an instructive approach for almost-uniform sampling for $SE(2)$ and establish for the first time what the fundamental

CHAPTER 4.

domains look like when the subgroup of $SE(2)$ is one of the five chiral wallpaper groups.

Part of the work in this chapter is published in [132, 133].

Chapter 5

Closed-Form Characterization of Minkowski Sum and Difference of Two Ellipsoids

In this chapter, a method is presented for parameterizing the exact boundaries of the Minkowski sum and difference of two ellipsoids. In general, the Minkowski sum and difference of two ellipsoids are not ellipsoids, but conventional approaches usually approximate them by families of inscribed and superscribed ellipsoids. The basic idea of our approach is to use a combination of affine transformations together with the analytic properties of offset surfaces to obtain exact closed-form parametric expressions for the boundaries. This approach can be used for any pair of triaxial ellipsoids in the case of Minkowski sums and differences when they exist. The exact closed-form characterization of the fundamental

CHAPTER 5.

results appears not to have been reported elsewhere in the literature. We also developed a new approach to quickly obtain ellipsoids that approximate the Minkowski sum and difference of two ellipsoids based on special properties of the Gaussian distribution. Both of these new results are compared with conventional approximations and with special cases known to have closed-form solutions. With our exact parameterization, the volumes of the Minkowski sum and difference of two ellipsoids can be calculated numerically, but in general, formulas for the volumes enclosed in these regions do not have exact closed-form expressions. We therefore develop closed-form bounds and approximations based on Steiner's Formula. These bounds and the numerically calculated ground truth are compared and analyzed for different planar elliptical and 3D ellipsoidal examples. Finally, we illustrate an application of these methods in the context of robot motion planning in which the volume of the space of collision-free motions is estimated in closed form for an ellipsoidal robot moving among ellipsoidal obstacles, all encapsulated in an ellipsoidal arena. This result builds on the Principal Kinematic Formula from Integral Geometry and a new formula, which we call the Kinematic Containment Formula.

5.1 Introduction

The Minkowski sum of two convex point sets (or bodies) centered at the origin, P_1 and P_2 in \mathbb{R}^n is denoted by $P_1 \oplus P_2$ and is defined as

$$P_1 \oplus P_2 = \{p_1 + p_2 \mid p_1 \in P_1, p_2 \in P_2\}. \quad (5.1)$$

CHAPTER 5.

The Minkowski difference between P_1 and P_2 , denoted by $P_1 \ominus P_2$, is defined as [134]

$$P_1 \ominus P_2 = \bigcap_{p_2 \in P_2} (P_1 + p_2), \quad (5.2)$$

Alternatively, by the definition of the Minkowski sum, we can define the Minkowski difference of two convex bodies as $P_1 \ominus P_2 = P'_1$, where $P_1 = P'_1 \oplus P_2$.

Minkowski operations are used in a wide range of applications such as robot motion planning [135], CAD/CAM, assembly planning [136] and computer-aided design [137]. For example, consider an obstacle P and a robot Q that moves by translation. We can choose a reference point attached to Q , then $P \oplus Q$ is the locus of positions of the reference point where $P \cap Q \neq \emptyset$. In the study of motion planning this sum is called a configuration space obstacle. Alternatively, if P is an arena in which the robot Q is moving, then $P \ominus Q$ is the locus of positions of the reference point where $P \cap Q = Q$, and represents the robot's collision-free configuration space with respect to the arena. While defining the Minkowski operations mathematically is easy, computing useful representations of Minkowski sums or differences can be difficult and computationally expensive, especially when the exact boundary of the geometry of these entities need to be represented explicitly. Much work has been done on obtaining boundaries of the Minkowski sums of two sets in two and three dimensions and on developing fast algorithms for computing Minkowski sums numerically [138]. The Minkowski sums of polygons (for two-dimensional cases) and polyhedra (for three-dimensional cases) have been extensively studied in the computational geometry [137, 139–142]. The related algorithms are mainly based on either the computation of the convolution of geometric boundaries [139], or polygon/polyhedra de-

CHAPTER 5.

compositions [137, 140–142]. Minkowski sums of curved regions/surfaces have also been studied (e.g. [143–146]).

Motivated by the robot motion planning problem using ellipsoidal bounding boxes, we present an approach to parameterizing the exact Minkowski sum and difference of two ellipsoids in closed forms. The Minkowski sum can be computed numerically by rapidly determining when two particular ellipsoidal bodies at given center positions and orientations intersect or not [147–150], or by ellipsoidal calculus [151–153]. However, we have not found the exact closed-form characterization of this fundamental result reported elsewhere in the literature. The basic idea of our approach is to use a combination of affine transformations together with the analytic properties of offset surfaces to obtain an exact closed-form parametric expression for the boundary. This approach can be used for any triaxial ellipsoids in the case of the Minkowski sums and differences. The Minkowski sum and difference of two ellipsoids are in general not ellipsoids, but can be approximated by ellipsoids. We also developed approaches to quickly obtain ellipsoidal approximations of the boundaries. One approach is based on affine transformations and approximation of offset curves. Another one uses Gaussian functions to model the interaction of ellipsoidal interaction and provides an upper bound for the Minkowski sum of two ellipsoids. Different than most of existing methods, our approaches are completely analytical and in closed form and therefore naturally provide an improved efficiency and a better robustness in many applications. The second half of the chapter studies volume bounds for the Minkowski sum of two ellipsoids. With our exact parameterization, they can be calculated numerically, however,

CHAPTER 5.

in general, formulas for the volumes enclosed in these regions do not have exact closed-form expressions. Based on Steiner's Formula, we develop an approach to provide an exact closed-form formula for the volume in axis-symmetric cases, and closed-form upper and lower bounds in more general triaxial cases.

In this chapter, we focus on the applications in 2- and 3-dimensions, but the approach is completely general and can be extended to n -dimensions. The remainder of this chapter is structured as follows. Sec. 5.2 formulates and solves the Minkowski sum and difference of two ellipsoids in closed forms. Sec. 5.3 provides two other approaches using ellipsoids to approximate the Minkowski sum and difference of two ellipsoids. Sec. 5.4 studies the volumes of the Minkowski sum of two ellipsoids. A Steiner's Formula-based approach is presented to provide closed-form bounds of those volumes. These bounds and the numerically calculated ground truth are compared and analyzed for different planar elliptical and three-dimensional ellipsoidal examples in Sec. 5.4.3. Then in Sec. 5.5 these results are applied to characterize the volume of free motion of an ellipsoidal robot moving among ellipsoidal obstacles in an ellipsoidal arena. We also introduce a new extension of the Principal Kinematic Formula. Finally, Sec. 5.6 presents our conclusions.

5.2 Closed-Form Characterization of Minkowski Operations of Two Ellipsoids

Our approach is to use a combination of affine transformations together with the analytic properties of offset surfaces to obtain an exact closed-form parametric expression for the boundaries of the Minkowski sum and difference of two ellipsoids.

5.2.1 The Minkowski sum of two ellipsoids

Let E_1 and E_2 be two triaxial ellipsoids with semi-axis lengths a_1, b_1, c_1 and a_2, b_2, c_2 , respectively. They are aligned in space relative to their principal axis frame with rotation angle θ_1, θ_2 and a rotation axis \mathbf{n}_1 and \mathbf{n}_2 . The Minkowski sum of E_1 and E_2 is $E_1 \oplus E_2 = \{\mathbf{x}_1 + \mathbf{x}_2 | \mathbf{x}_1 \in E_1, \mathbf{x}_2 \in E_2\}$. Let E_2 move around E_1 and attach a reference point in the center of E_2 . Then the boundary of $E_1 \oplus E_2$ becomes the locus of positions of the reference point of E_2 where $P \cap Q \neq \emptyset$. The basic idea of our approach is to apply affine transformations to E_1 and E_2 that result in shrinking E_2 into a sphere of the radius r , and calculating an offset surface of E_1 with radius r . We then stretch the offset surface using the opposite of the previous shrinking ratio. For a graphical explanation, the algorithm is illustrated with a planar example in Fig. 5.1. In Fig. 5.1 (a), the boundary of the Minkowski sum of E_1 and E_2 is constructed by the center of E_2 when it is touching E_1 . It forms a deformed offset curve/surface of the motionless ellipsoid with variable offset distances. To characterize the

CHAPTER 5.

boundary of $E_1 \oplus E_2$, first as shown in Fig. 5.1 (b) we shrink both ellipsoids together until E_2 changes to a circle/sphere with radius r . Here, r is chosen as the smallest semi-axis length i.e., $r = \min\{a_2, b_2, c_2\}$. Therefore, the boundary of $E_1 \oplus E_2$ in this case becomes an offset curve/surface. After this affine operation E_1 still remains as an ellipsoid but with changed semi-axis lengths a'_1, b'_1, c'_1 and rotation axis and rotation angle \mathbf{n}'_1, θ'_1 .

The “shrinking” operation on E_1 can be represented as

$$\mathbf{x}'_1 = R(\mathbf{n}_2, \theta_2) \Delta^{-1}(a_2/r, b_2/r, c_2/r) R(\mathbf{n}_2, -\theta_2) \mathbf{x}_1, \quad (5.3)$$

where \mathbf{x}_1 and \mathbf{x}'_1 specify the coordinates of the original and shrunk versions of E_1 , respectively. $R \in SO(3)$ is the rotation matrix, i.e., from Euler’s Theorem,

$$R(\mathbf{n}, \theta) = \exp(\theta \hat{\mathbf{n}}), \quad (5.4)$$

where \mathbf{n} is a unit vector, $\hat{\mathbf{n}}$ represents the skew-symmetric matrix of the vector \mathbf{n} such that $\hat{\mathbf{n}} \mathbf{x} = \mathbf{n} \times \mathbf{x}$, and $\exp(\cdot)$ is the matrix exponential. $\Delta(a_2, b_2, c_2)$ is defined as the diagonal matrix, i.e.,

$$\Delta(a_2, b_2, c_2) = \begin{pmatrix} a_2 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & c_2 \end{pmatrix}, \quad (5.5)$$

and

$$\Delta^{-1}(a_2, b_2, c_2) = \begin{pmatrix} 1/a_2 & 0 & 0 \\ 0 & 1/b_2 & 0 \\ 0 & 0 & 1/c_2 \end{pmatrix}. \quad (5.6)$$

CHAPTER 5.

Then \mathbf{x}_1 can be represented as

$$\mathbf{x}_1 = R(\mathbf{n}_2, \theta_2)\Delta(a_2/r, b_2/r, c_2/r)R(\mathbf{n}_2, -\theta_2)\mathbf{x}'_1, \quad (5.7)$$

The implicit expression for the original version of E_1 is

$$\mathbf{x}'_1{}^T R(\mathbf{n}_1, \theta_1)\Delta^{-1}(a_1^2, b_1^2, c_1^2)R(\mathbf{n}_1, -\theta_1)\mathbf{x}_1 = 1 \quad (5.8)$$

By substituting (5.7) into (5.8), we can get the implicit expression of the shrunk version of E_1 as,

$$\begin{aligned} & [R(\mathbf{n}_2, \theta_2)\Delta(a_2/r, b_2/r, c_2/r)R(\mathbf{n}_2, -\theta_2)\mathbf{x}'_1]^T \\ & R(\mathbf{n}_1, \theta_1)\Delta^{-1}(a_1^2, b_1^2, c_1^2)R(\mathbf{n}_1, -\theta_1) \\ & [R(\mathbf{n}_2, \theta_2)\Delta(a_2/r, b_2/r, c_2/r)R(\mathbf{n}_2, -\theta_2)\mathbf{x}'_1] = 1, \\ \Rightarrow & (\mathbf{x}'_1)^T R(\mathbf{n}_2, \theta_2)\Delta(a_2/r, b_2/r, c_2/r)R(\mathbf{n}_2, -\theta_2) \\ & R(\mathbf{n}_1, \theta_1)\Delta^{-1}(a_1^2, b_1^2, c_1^2)R(\mathbf{n}_1, -\theta_1), \\ & R(\mathbf{n}_2, \theta_2)\Delta(a_2/r, b_2/r, c_2/r)R(\mathbf{n}_2, -\theta_2)\mathbf{x}'_1 = 1, \\ \Rightarrow & (\mathbf{x}'_1)^T R(\mathbf{n}'_1, \theta'_1)\Delta^{-1}(a'^2_1, b'^2_1, c'^2_1)R(\mathbf{n}'_1, -\theta'_1)\mathbf{x}'_1 = 1, \end{aligned} \quad (5.9)$$

where a'_1, b'_1, c'_1 are the changed semi-axis lengths, and \mathbf{n}'_1 and θ'_1 are the changed rotation axis and angle of E_1 , respectively. Suppose $\theta'_1=0$, the parametric equation of the shrunk motionless ellipsoid

$$\phi(\mathbf{x}'_1) = \frac{x'^2_1}{a'^2_1} + \frac{y'^2_1}{b'^2_1} + \frac{z'^2_1}{c'^2_1} - 1 = 0 \quad (5.10)$$

CHAPTER 5.

can be written as

$$x'_1(u, v) = a'_1 \cos u \sin v, \quad (5.11)$$

$$y'_1(u, v) = b'_1 \sin u \sin v,$$

$$z'_1(u, v) = c'_1 \cos v,$$

where $u \in [0, 2\pi]$ and $v \in [0, \pi]$.

The offset surface \mathbf{x}_{ofs} of a differentiable surface \mathbf{x} with the radius r is defined as

$$\mathbf{x}_{ofs} = \mathbf{x} + r\mathbf{n}, \quad (5.12)$$

where \mathbf{n} is the unit surface normal. In the case of the ellipsoidal surface in (5.10),

$$\mathbf{n} = \frac{\nabla\phi(\mathbf{x}'_1)}{\|\nabla\phi(\mathbf{x}'_1)\|} = [2x'_1/a_1'^2, 2y'_1/b_1'^2, 2z'_1/c_1'^2]^T / \|\nabla\phi(\mathbf{x}'_1)\|, \quad (5.13)$$

where $\mathbf{x}'_1 = [x'_1, y'_1, z'_1]^T$, and

$$\|\nabla\phi(\mathbf{x}'_1)\| = \left[\left(\frac{2x'_1}{a_1'^2} \right)^2 + \left(\frac{2y'_1}{b_1'^2} \right)^2 + \left(\frac{2z'_1}{c_1'^2} \right)^2 \right]^{1/2}. \quad (5.14)$$

Therefore, based on (5.11), the offset surface of the ellipsoid in (5.10), i.e., the shrunk version of E_1 , can be parametrized as

$$\begin{aligned} x'_{ofs}(u, v) &= x'_1(u, v) + \frac{2rx'_1(u, v)}{a_1'^2 \|\nabla\phi((\mathbf{x}'_1)(u, v))\|}, \\ y'_{ofs}(u, v) &= y'_1(u, v) + \frac{2ry'_1(u, v)}{b_1'^2 \|\nabla\phi((\mathbf{x}'_1)(u, v))\|}, \\ z'_{ofs}(u, v) &= z'_1(u, v) + \frac{2rz'_1(u, v)}{c_1'^2 \|\nabla\phi((\mathbf{x}'_1)(u, v))\|}, \end{aligned} \quad (5.15)$$

In general $\theta'_1 \neq 0$, and we need to rotate the offset curve

$$\mathbf{x}_{ofs} = [x_{ofs}, y_{ofs}, z_{ofs}]^T$$

to the direction of the shrunk version of E_1 by multiplying the rotation matrix $R(\mathbf{n}'_1, \theta'_1)$. After that, we just need to stretch it back until the shrunk version of E_1 becomes to the original one again (see Fig. 5.1 (c)), and after the “stretching” operation, the exact boundary of $E_1 \oplus E_2$ can be finally represented in closed form as

$$\mathbf{x}_{eb} = TR(\mathbf{n}'_1, \theta'_1)\mathbf{x}_{ofs}, \quad (5.16)$$

where

$$T = R(\mathbf{n}_2, \theta_2)\Delta(a_2/r, b_2/r, c_2/r)R(\mathbf{n}_2, -\theta_2). \quad (5.17)$$

5.2.2 The Minkowski difference of two ellipsoids

In the previous section, we characterize the Minkowski sum of two ellipsoids E_1 and E_2 . Based on the same of idea, we can also characterize their Minkowski difference, i.e., $E_1 \ominus E_2 = \{\mathbf{y} | \mathbf{y} + \mathbf{x}_2 \in E_1, \mathbf{x}_2 \in E_2\}$. Compared to $E_1 \oplus E_2$, in the parametric representation, the only difference is that the offset curve/surface is in the direction of inward-facing normal of the ellipsoid rather than the outward (see Fig. 5.2). In other words, with the other parts of the derivation remain the same, the plus sign need to be changed to a minus sign in (5.15), i.e.,

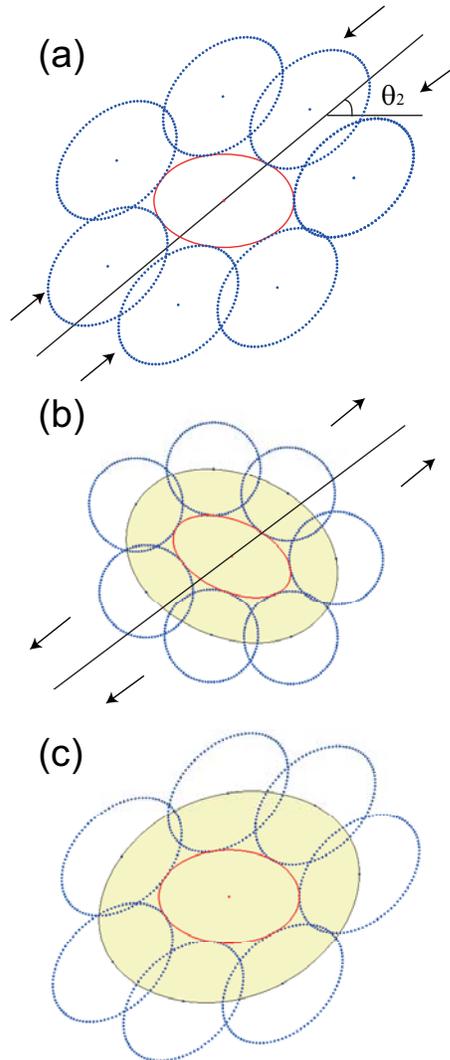


Figure 5.1: The algorithm for obtaining the parametric representation of the boundary of the Minkowski sum of two ellipses E_1 and E_2 . In (a), both ellipses are shrunk in the direction with the rotation angle θ_2 (see (5.3)) until E_2 becomes a circle. After the shrinking process, E_1 remains elliptical but with different semi-axis lengths and rotation angle, whereas E_2 becomes a circle (see (b)). The Minkowski sum in this transformed space is then an offset surface. In (c), everything is stretched back in the same direction until the surrounding circle becomes the original version of E_2 again. The shaded region in (c) represents $E_1 \oplus E_2$. We note that the shaded regions in (b) and (c) that may appear to be elliptical are actually not ellipses, and the amount that they deviate from being ellipses depends on the eccentricity of the elliptical bodies.

$$\begin{aligned}
x'_{ofs}(u, v) &= x'_1(u, v) - \frac{2rx'_1(u, v)}{a_1^2 \|\nabla\phi((\mathbf{x}'_1)(u, v))\|}, \\
y'_{ofs}(u, v) &= y'_1(u, v) - \frac{2ry'_1(u, v)}{b_1^2 \|\nabla\phi((\mathbf{x}'_1)(u, v))\|}, \\
z'_{ofs}(u, v) &= z'_1(u, v) - \frac{2rz'_1(u, v)}{c_1^2 \|\nabla\phi((\mathbf{x}'_1)(u, v))\|},
\end{aligned} \tag{5.18}$$

5.3 Approximation of the Minkowski Operations of Two Ellipsoids

5.3.1 Ellipsoidal approximation of the offset surfaces

The algorithms we developed in Sec. 5.2 also provides us a natural way to derive a closed-form approximation of $E_1 \oplus E_2$ and $E_1 \ominus E_2$. Alternatively, noticing that the offset surface of an ellipsoid with the length r can be approximated by an ellipsoid with semi-axis lengths extended/shrunk by the radius r (for $r \geq 0$), so as in (5.15), instead of using the offset surfaces, we can use an ellipsoid with extended/shrunk semi-axis lengths to approximate $E_1 \oplus E_2$ and $E_1 \ominus E_2$. The implicit representation of the extended/shrunk ellipsoid can be written as,

$$\mathbf{x}_{ea}^T R(\mathbf{n}'_1, \theta'_1) \Delta_{\pm}^{-1} R^T(\mathbf{n}_1, -\theta'_1) \mathbf{x}_{ea} = 1, \tag{5.19}$$

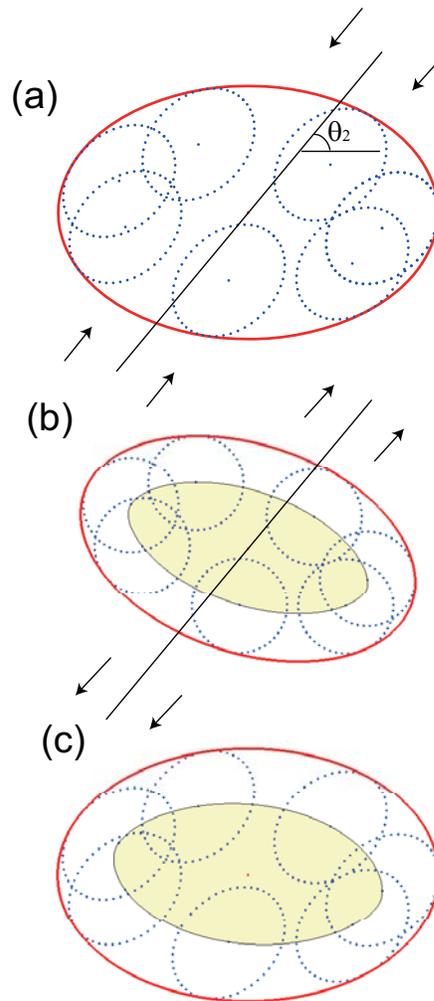


Figure 5.2: The algorithm for obtaining the parametric representation of the boundary of the Minkowski difference of two ellipses E_1 and E_2 . This presumes that the radius of curvature of the resulting circle/sphere is smaller than the smallest semi-axis length of the bounding shrunk ellipsoid. The algorithm follows the same steps as Fig. 5.1. The shaded region in (c) represents $E_1 \ominus E_2$.

CHAPTER 5.

where

$$\Delta_{\pm}^{-1} = \Delta^{-1} \left((a'_1 \pm r)^2, (b'_1 \pm r)^2, (c'_1 \pm r)^2 \right), \quad (5.20)$$

and “+” and “−” correspond to the extended and shrunk semi-axis lengths in $E_1 \oplus E_2$ and $E_1 \ominus E_2$, respectively.

Using the same stretching process in (5.16), we can have the implicit expression of the ellipsoidal approximations of $E_1 \oplus E_2$ and $E_1 \ominus E_2$

$$\mathbf{x}_{ea}^T T R(\mathbf{n}'_1, \theta'_1) \Delta_{\pm}^{-1} R^T(\mathbf{n}_1, -\theta'_1) T^T \mathbf{x}_{ea} = 1.$$

Since the volume of a triaxial ellipsoid in three-dimensional space with semi-axis lengths a, b, c is simply

$$V = \frac{4}{3} \pi abc \quad (5.21)$$

and by the fact that

$$V(A \cdot C) = |\det A| \cdot V(C), \quad (5.22)$$

we can compute the approximation (5.21) of the volume of Minkowski sums as

$$V = \frac{4}{3} \pi |\det A| (a'_1 + r)(b'_1 + r)(c'_1 + r), \quad (5.23)$$

where

$$A = R(\mathbf{n}_2, \theta_2) \Delta(r/a_1, r/b_1, r/c_1) R(\mathbf{n}_2, -\theta_2) R(\mathbf{n}'_1, \theta'_1). \quad (5.24)$$

5.3.2 Ellipsoidal approximation based on Gaussian functions

The upper bound of the Minkowski sum of two ellipsoids can be characterized by the covariance matrix of Gaussian functions that approximates the interactions of ellipsoids E_1 and E_2 . First, the generalized ellipsoid equation can be written as

$$(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \leq 1, \quad (5.25)$$

where

$$\boldsymbol{\Sigma}_i^{-1} = R\Delta(1/a^2, 1/b^2, 1/c^2)R^T. \quad (5.26)$$

Then a Gaussian function normalized to a unit peak height, rather than as a probability density function, can be used to represent this ellipsoid, i.e.,

$$f(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\right]. \quad (5.27)$$

The height of this ellipsoidal-shaped contour plane of the corresponding Gaussian is $e^{-1/2}$. When the two ellipsoids “kiss” each other, the product of the corresponding Gaussians is

$$f(\mathbf{x}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) f(\mathbf{x}, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) = e^{-1/2} e^{-1/2} = e^{-1}, \quad (5.28)$$

when they collide with each other, the product is greater than e^{-1} and if there is no overlap between them, the product is less than e^{-1} .

We can write the product of two Gaussians in the form of $C(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) f(\mathbf{x}, \boldsymbol{\mu}', \boldsymbol{\Sigma}')$,

CHAPTER 5.

where $C(\mu_1, \mu_2, \Sigma_1, \Sigma_2)$ is a constant and $f(\mathbf{x}, \mu', \Sigma')$ is another Gaussian function (normalized to a unit peak height).

To simplify the problem, let $\mu_1 = 0$, then

$$\begin{aligned}
 & f(\mathbf{x}, 0, \Sigma_1) f(\mathbf{x}, \mu_2, \Sigma_2) & (5.29) \\
 & = \exp\left[-\frac{1}{2}\mathbf{x}^T \Sigma_1^{-1} \mathbf{x} - \frac{1}{2}(\mathbf{x} - \mu_2)^T \Sigma_2^{-1} (\mathbf{x} - \mu_2)\right] \\
 & = \exp\left[-\frac{1}{2}(\mathbf{x}^T (\Sigma_1^{-1} + \Sigma_2^{-1}) \mathbf{x} - 2\mathbf{x}^T \Sigma_2^{-1} \mu_2 + \mu_2^T \Sigma_2^{-1} \mu_2)\right] \\
 & = C(\mu_2, \Sigma_1, \Sigma_2) f(\mathbf{x}, \mu', \Sigma'),
 \end{aligned}$$

where

$$C(\mu_2, \Sigma_1, \Sigma_2) = \exp\left[-\frac{1}{2}\mu_2^T (\Sigma_2^{-1} - \Sigma_2^{-1} \Sigma' \Sigma_2^{-1}) \mu_2\right], \quad (5.30)$$

$$\Sigma' = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}, \quad (5.31)$$

$$\mu' = \Sigma' \Sigma_2^{-1} \mu_2 = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \Sigma_2^{-1} \mu_2. \quad (5.32)$$

Since $f(\mathbf{x}, \mu', \Sigma') \leq 1$, $C(\mu_2, \Sigma_1, \Sigma_2) f(\mathbf{x}, \mu', \Sigma') \geq e^{-1}$ when two ellipsoids intersect, then

$$C(\mu_2, \Sigma_1, \Sigma_2) \geq e^{-1} \quad (5.33)$$

$$\Rightarrow \mu_2^T \Sigma'' \mu_2 \leq 1.$$

where $\Sigma'' = \frac{1}{2}(\Sigma_2^{-1} - \Sigma_2^{-1} \Sigma' \Sigma_2^{-1})$. The ellipsoidal-shaped region $\mu_2^T \Sigma'' \mu_2 \leq 1$ provides an upper bound for the Minkowski sum of the two ellipsoids.

Figs. 5.3 and 5.4 illustrate a planar and a spatial example of approximations on the Minkowski-sum boundary of two ellipses/ellipsoids. We note that for the planar case,

the elliptical approximations discussed in Secs. 5.3.1 and 5.3.1 provide lower and upper bounds of the Minkowski-sum boundary. However, these approximations may no longer be bounds in spatial cases.

5.4 The Volume of the Minkowski Sum of Two Ellipsoids

Here we show that using our methods it is possible to approximate the volume of the Minkowski sum of two arbitrary ellipsoids with lower bounds that are tighter than the *Brunn-Minkowski* inequality

$$V(E_1 \oplus E_2)^{\frac{1}{n}} \geq V(E_1)^{\frac{1}{n}} + V(E_2)^{\frac{1}{n}}. \quad (5.34)$$

Moreover, using the similar methods, we generate upper bounds. The bounds that we develop in this section for volumes of Minkowski sums use properties of offset curves and surfaces and Steiner's formula. We begin by reviewing these concepts.

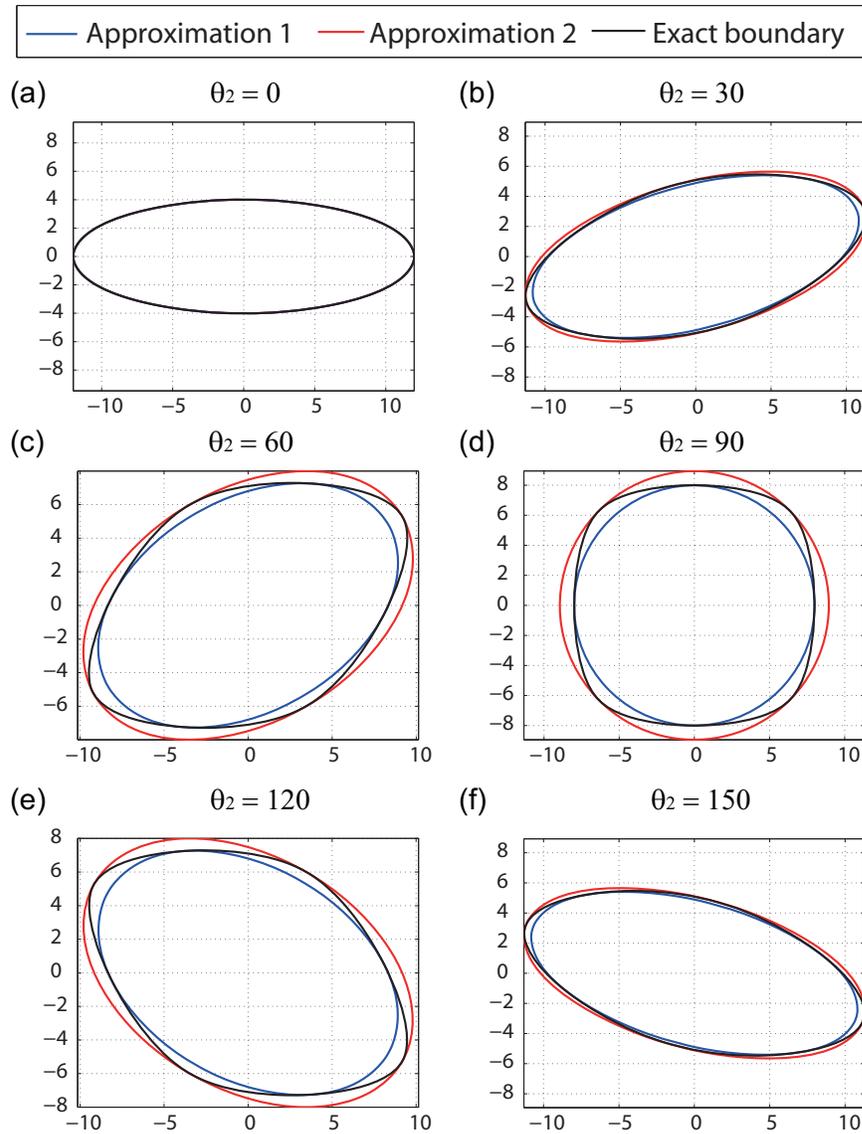
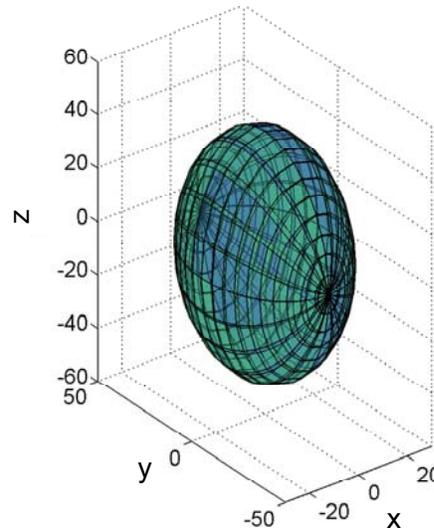


Figure 5.3: An planar example: the exact boundary (black), approximated elliptical boundary 1, based on the approximation of the offset curves (blue) and approximated elliptical boundary 2, based on Gaussian functions (red) of the Minkowski sum of two ellipses. The semi-axis lengths of E_1 and E_2 are 6, 2, and 10, 2, respectively. The six subplots correspond to six different angles (θ_2) of E_2 .

(a) Approximation 1 (blue) VS Eaxt Boundary (green)



(b) Approximation 2 (red) VS Eaxt Boundary (green)

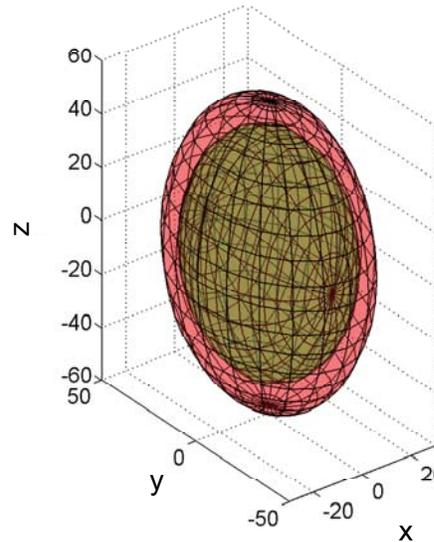


Figure 5.4: A 3D ellipsoidal example: the exact boundary (green), approximated ellipsoidal boundary 1, based on the approximation of the offset surfaces (blue) and approximated ellipsoidal boundary 2, based on Gaussian functions (red) of the Minkowski sum of two ellipsoids. The semi-axis lengths of E_1 and E_2 are 20, 30, 40, and 6, 2, 2, respectively. The ZXZ Euler angles of E_1 and E_2 are $\alpha_1 = 0, \beta_1 = 0, \gamma_1 = 0$ and $\alpha_2 = 0, \beta_2 = 0, \gamma_2 = 0$, respectively.

5.4.1 Bounds on the Minkowski sum volume of two ellipsoids via Steiner's formula (for planar cases)

Given a parametric convex curve in the plane, $\mathbf{x}(s)$, a parallel curve can be constructed external to it as

$$\mathbf{o}(s; r) = \hat{\mathbf{o}}(\mathbf{x}(s); r) \doteq \mathbf{x}(s) + r\mathbf{n}(s) \quad (5.35)$$

where $\mathbf{n}(s)$ is the planar outward-pointing unit normal to the curve at each point. For each fixed value of r , the curve $\mathbf{o}(s; r)$ is called an *offset curve*.

For convenience, the curve parameter s is usually chosen to be arc length. Then it can be shown that

$$\frac{d\mathbf{o}(s; r)}{ds} = [1 + r\kappa(s)]\mathbf{u}(s) \quad (5.36)$$

where

$$\mathbf{u}(s) = \frac{d\mathbf{x}(s)}{ds}, \quad (5.37)$$

$$\text{and } \kappa(s) = \left(\frac{d^2\mathbf{x}(s)}{ds^2} \cdot \frac{d^2\mathbf{x}(s)}{ds^2} \right)^{\frac{1}{2}}$$

are respectively the unit tangent and the unsigned curvature of the original curve. Even though s is not the arc length of the offset curve $\mathbf{o}(s; r)$, the unit tangent to $\mathbf{o}(s; r)$ is also $\mathbf{u}(s)$. And the unit normal to $\mathbf{o}(s; r)$ is also $\mathbf{n}(s)$. Therefore, since $\mathbf{x}(s)$ and $\mathbf{o}(s; r)$ have the same normal and tangent, it follows that taking the offset curve of an offset curve results in an offset curve of the original curve:

$$\hat{\mathbf{o}}(\hat{\mathbf{o}}(\mathbf{x}(s); r_1); r_2) = \hat{\mathbf{o}}(\mathbf{x}(s); r_1 + r_2). \quad (5.38)$$

CHAPTER 5.

The area of the strip contained between the original closed curve bounding the convex planar body and the offset curve can be computed as

$$\begin{aligned}
 V &= \int_0^{r_0} \int_0^L |\det[d\mathbf{o}/ds, d\mathbf{o}/dr]| ds dr & (5.39) \\
 &= \int_0^{r_0} \int_0^L [1 + \kappa(s)r] |\det[\mathbf{u}, \mathbf{n}]| ds dr \\
 &= \int_0^{r_0} \int_0^L [1 + \kappa(s)r] ds dr \\
 &= r_0 L + \frac{r_0^2}{2} \int_0^L \kappa(s) ds
 \end{aligned}$$

And so, the area within the offset curve is the sum of the area inside of the original curve with the two terms given above. (The perimeter times the offset, and a term quadratic in the offset times the total curvature). However, it has been known for centuries that

$$\int_0^L \kappa(s) ds = 2\pi \tag{5.40}$$

and so if we can compute the perimeter, we can compute the volume enclosed by the offset curve.

Therefore, for planar cases, Steiner's Formula becomes ¹

$$V(B_r) = V(B) + rL(\partial B) + \pi r^2, \tag{5.41}$$

where $L(\partial B)$ represents the perimeter of the ellipse. The perimeter of an ellipse with semi-axis lengths a and b with $a \leq b$ can be written exactly as

$$L(a, b) = 4b E\left(\frac{\pi}{2}, \sqrt{1 - \frac{a^2}{b^2}}\right) \tag{5.42}$$

¹Since area takes the place of volume in 2D problems, and we retain the symbol V when referring to area.

where $E(\varphi, m)$ is the incomplete elliptic integral of the second kind. Using this, the exact area contained inside of the offset curve can be obtained, and the exact area of the Minkowski sum of two ellipses results. Nevertheless, it can be useful to evaluate lower and upper bounds using elementary functions.

The perimeter of an ellipse can be approximated by Legendre's exact expansion [154],

$$L(a, b) = 2\pi a \left[1 - \frac{w^2}{4} - \frac{3w^4}{64} - (2\nu - 1)^{-1} \binom{2\nu}{\nu} \left(\frac{w}{4}\right)^{2\nu} - \dots \right], \quad (5.43)$$

where w is the eccentricity of the ellipse, i.e., with $a \geq b$,

$$w = \sqrt{1 - \frac{b^2}{a^2}}. \quad (5.44)$$

Therefore, we can use the first few terms as an upper bound, for example

$$L_{ub} = 2\pi a \left(1 - \frac{w^4}{4} - \frac{3w^2}{64} \right). \quad (5.45)$$

Also, by the isoperimetric inequality [155], we have a lower bound of L as

$$L_{lb} = \sqrt{4\pi^2 ab + 4\pi(a - b)^2}. \quad (5.46)$$

5.4.2 Bounds on the Minkowski sum volume of two ellipsoids via Steiner's formula (for spatial cases)

For a finite body B with volume $V(B)$ enclosed by a compact surface ∂B , *Steiner's formula* calculates the volume enclosed by the surface offset by an amount r from ∂B

CHAPTER 5.

[156]:

$$V(B_r) = V(B) + rF(\partial B) + r^2M(\partial B) + \frac{r^3}{3}K(\partial B). \quad (5.47)$$

Here $F(\partial B)$ is the area of the bounding surface, $M(\partial B)$ is the mean curvature integrated over the bounding surface, and $K(\partial B)$ is the Gaussian curvature integrated over the bounding surface. From the Gauss-Bonnet theorem applied to the surface bounding a simply-connected body (as must be the case for a convex body),

$$K(\partial B) = 4\pi. \quad (5.48)$$

And so, if the surface area and integral of mean curvature can be computed, we can exactly compute the volume of the offset surface.

This, together with (5.22) and our construction of the Minkowski sum of ellipsoids by the application of appropriate linear transformations resulting in offset surfaces of ellipsoids, allows us to bound the volume of the Minkowski sum of ellipsoids in closed form. Here we seek to bound these quantities from below and above, thereby bounding the volume of the offset of an ellipse, and from our previous construction, bounding the volume of the Minkowski sum of two arbitrary ellipsoids at arbitrary orientations.

In the spatial case, both the surface area and mean curvature must either be computed, bounded, or approximated.

5.4.2.1 Bounds on the total mean curvature of a triaxial ellipsoid

Exact formulas are not known to us for the total mean curvature M , for a triaxial ellipsoid. But several approaches to bounding this quantity are possible.

Given the parameterized equation of a triaxial ellipsoid

$$\mathbf{x}(\phi, \theta) = \begin{pmatrix} a \cos \phi \sin \theta \\ b \sin \phi \sin \theta \\ c \cos \theta \end{pmatrix}$$

the outward pointing normal is

$$\begin{aligned} \mathbf{n}(\phi, \theta) &= \left(\frac{\partial \mathbf{x}}{\partial \theta} \times \frac{\partial \mathbf{x}}{\partial \phi} \right) \cdot \left\| \frac{\partial \mathbf{x}}{\partial \theta} \times \frac{\partial \mathbf{x}}{\partial \phi} \right\|^{-1} \\ &= \begin{pmatrix} bc \cos \phi \sin \theta \\ ac \sin \phi \sin \theta \\ ab \cos \theta \end{pmatrix} \cdot (b^2 c^2 \cos^2 \phi \sin^2 \theta + a^2 c^2 \sin^2 \phi \sin^2 \theta + a^2 b^2 \cos^2 \theta)^{-1/2} \end{aligned}$$

Therefore,

$$\mathbf{x} \cdot \mathbf{n} = abc \cdot (b^2 c^2 \cos^2 \phi \sin^2 \theta + a^2 c^2 \sin^2 \phi \sin^2 \theta + a^2 b^2 \cos^2 \theta)^{-1/2}$$

and the element of surface area is

$$\begin{aligned} dS &= \left\| \frac{\partial \mathbf{x}}{\partial \theta} \times \frac{\partial \mathbf{x}}{\partial \phi} \right\| d\phi d\theta \\ &= (b^2 c^2 \cos^2 \phi \sin^2 \theta + a^2 c^2 \sin^2 \phi \sin^2 \theta + a^2 b^2 \cos^2 \theta)^{1/2} \sin \theta d\phi d\theta. \end{aligned} \tag{5.49}$$

CHAPTER 5.

In general, the mean curvature can be computed using the formula

$$\begin{aligned} m &= \frac{\|\nabla\phi\|^2 \operatorname{tr}(\nabla\nabla^T\phi) - (\nabla^T\phi)(\nabla\nabla^T\phi)(\nabla\phi)}{2\|\nabla\phi\|^3} \\ &= \nabla \cdot \left(\frac{\nabla\phi}{\|\nabla\phi\|} \right). \end{aligned} \quad (5.50)$$

For the triaxial ellipsoid

$$\phi(\mathbf{x}) \doteq \mathbf{x}^T A \mathbf{x} - 1 = 0 \quad (5.51)$$

with

$$A = \Delta^{-1}(a^2, b^2, c^2), \quad (5.52)$$

we evaluate at $\mathbf{x} = \mathbf{x}(\phi, \theta)$ (using the same parametric equation above), which gives

$$\begin{aligned} m(\phi, \theta) &= \frac{(\mathbf{x}^T A^2 \mathbf{x}) \operatorname{tr}(A) - \mathbf{x}^T A^3 \mathbf{x}}{2 \cdot (\mathbf{x}^T A^2 \mathbf{x})^{3/2}} \\ &= \frac{abc [(a^2 + b^2) \cos^2 \theta + ((a^2 + c^2) \sin^2 \phi + (b^2 + c^2) \cos^2 \phi) \sin^2 \theta]}{2(a^2 b^2 \cos^2 \theta + c^2(b^2 \cos^2 \phi + a^2 \sin^2 \phi) \sin^2 \theta)^{3/2}} \end{aligned} \quad (5.53)$$

The total mean curvature of a triaxial ellipsoid is

$$\begin{aligned} M &= \int_{\partial B} m dS \\ &= \int_0^\pi \int_0^{2\pi} \frac{abc [(a^2 + b^2) \cos^2 \theta + ((a^2 + c^2) \sin^2 \phi + (b^2 + c^2) \cos^2 \phi) \sin^2 \theta]}{2(a^2 b^2 \cos^2 \theta + c^2(b^2 \cos^2 \phi + a^2 \sin^2 \phi) \sin^2 \theta)} \sin \theta d\phi d\theta. \end{aligned} \quad (5.54)$$

We note that the fractional power disappeared due to the product of $m \cdot dS$.

One approach to obtain bounds for M is to tackle the problem of by integrating M over ϕ first

CHAPTER 5.

In order to simplify the discussion, we denote

$$k_0 = \frac{abc}{2}, \quad (5.55)$$

$$k_1(\theta) = [(a^2 + c^2) \sin^2 \theta + (a^2 + b^2) \cos^2 \theta] \sin \theta,$$

$$k_2(\theta) = [(b^2 + c^2) \sin^2 \theta + (a^2 + b^2) \cos^2 \theta] \sin \theta,$$

$$h_1(\theta) = (a^2 b^2 \cos^2 \theta + a^2 c^2 \sin^2 \theta)^{1/2},$$

$$h_2(\theta) = (a^2 b^2 \cos^2 \theta + b^2 c^2 \sin^2 \theta)^{1/2}.$$

and M in (5.54) can be rewritten as

$$M = k_0 \int_0^\pi \int_0^{2\pi} \frac{k_1(\theta) \sin^2 \phi + k_2(\theta) \cos^2 \phi}{h_1^2(\theta) \sin^2 \phi + h_2^2(\theta) \cos^2 \phi} d\phi d\theta. \quad (5.56)$$

We integrate M over ϕ first

$$\begin{aligned} & \int_0^{2\pi} \frac{k_1(\theta) \sin^2 \phi + k_2(\theta) \cos^2 \phi}{h_1^2(\theta) \sin^2 \phi + h_2^2(\theta) \cos^2 \phi} d\phi \\ &= 2\pi \left(\frac{k_1(\theta)}{h_1(\theta)(h_1(\theta) + h_2(\theta))} + \frac{k_2(\theta)}{h_2(\theta)(h_1(\theta) + h_2(\theta))} \right). \end{aligned} \quad (5.57)$$

Since

$$\begin{aligned} \int_0^{\pi/2} \frac{\cos^2 x dx}{\alpha^2 \sin^2 x + \beta^2 \cos^2 x} &= \frac{\pi}{2\beta(\alpha + \beta)}, \\ \int_0^{\pi/2} \frac{\sin^2 x dx}{\alpha^2 \sin^2 x + \beta^2 \cos^2 x} &= \frac{\pi}{2\alpha(\alpha + \beta)}, \end{aligned}$$

moreover, a general principle is that if

$$A = \int_0^{\pi/2} f(\cos^2 x, \sin^2 x) dx \text{ and } B = \int_0^{\pi/2} f(\sin^2 x, \cos^2 x) dx$$

then

$$\int_0^\pi f(\cos^2 x, \sin^2 x) dx = A + B \text{ and } \int_0^{2\pi} f(\cos^2 x, \sin^2 x) dx = 2(A + B),$$

CHAPTER 5.

and so

$$\int_0^{2\pi} \frac{\cos^2 x dx}{\alpha^2 \sin^2 x + \beta^2 \cos^2 x} = \frac{2\pi}{\beta(\alpha + \beta)}$$

,

$$\int_0^{2\pi} \frac{\sin^2 x dx}{\alpha^2 \sin^2 x + \beta^2 \cos^2 x} = \frac{2\pi}{\alpha(\alpha + \beta)}$$

Then M becomes

$$M = 2k_0\pi \int_0^\pi \frac{k_1(\theta)}{h_1(\theta)(h_1(\theta) + h_2(\theta))} + \frac{k_2(\theta)}{h_2(\theta)(h_1(\theta) + h_2(\theta))} d\theta. \quad (5.58)$$

in which

$$\begin{aligned} & \int_0^\pi \frac{k_1(\theta)}{h_1(\theta)(h_1(\theta) + h_2(\theta))} d\theta \\ &= \int_{-1}^1 \frac{\alpha_1 t^2 + \beta_1}{\sqrt{\alpha_2 t^2 + \beta_2} (\sqrt{\alpha_2 t^2 + \beta_2} + \sqrt{\alpha_3 t^2 + \beta_3})} dt. \end{aligned} \quad (5.59)$$

where

$$\alpha_1 = b^2 - c^2, \quad (5.60)$$

$$\alpha_2 = a^2(b^2 - c^2),$$

$$\alpha_3 = b^2(a^2 - c^2),$$

$$\beta_1 = a^2 + c^2,$$

$$\beta_2 = a^2 c^2,$$

$$\beta_3 = b^2 c^2.$$

We can bound the term $\sqrt{(\alpha_2 t^2 + \beta_2)(\alpha_3 t^2 + \beta_3)}$ as

$$\alpha' t^2 + \beta' \leq \sqrt{(\alpha_2 t^2 + \beta_2)(\alpha_3 t^2 + \beta_3)} \leq \alpha'' t^2 + \beta'' \quad (5.61)$$

CHAPTER 5.

and in this way, replace denominators with $\alpha't^2 + \beta'$ and $\alpha''t^2 + \beta''$, and bounding integrals that can be computed in closed form.

To get (α', β') and (α'', β'') , first we expand out

$$(\alpha_2 t^2 + \beta_2)(\alpha_3 t^2 + \beta_3) = (\alpha_2 \alpha_3) t^4 + (\alpha_2 \beta_3 + \alpha_3 \beta_2) t^2 + \beta_2 \beta_3.$$

Next, compare with a candidate perfect square

$$(\alpha t^2 + \beta)^2 = \alpha^2 t^4 + 2\alpha\beta t^2 + \beta^2.$$

We have complete control over defining $\alpha, \beta \in \mathbb{R}_{\geq 0}$, and want to choose so that one of the results $(\alpha t^2 + \beta)^2 \leq (\alpha_2 t^2 + \beta_2)(\alpha_3 t^2 + \beta_3)$ or $(\alpha t^2 + \beta)^2 \geq (\alpha_2 t^2 + \beta_2)(\alpha_3 t^2 + \beta_3)$ holds, and is as tight as we can make it. If we choose

$$\beta^2 \doteq \beta_2 \beta_3 \quad \text{and} \quad 2\alpha\beta \doteq \alpha_2 \beta_3 + \alpha_3 \beta_2$$

then the difference

$$(\alpha t^2 + \beta)^2 - (\alpha_2 t^2 + \beta_2)(\alpha_3 t^2 + \beta_3) = t^4 \frac{(\beta_2 \alpha_3 - \beta_3 \alpha_2)^2}{4\beta_2 \beta_3} \geq 0.$$

And since t^4 is very small on much of the range $-1 \leq t \leq 1$, I would expect that using this will result in a good upper bound of the form

$$\sqrt{(\alpha_2 t^2 + \beta_2)(\alpha_3 t^2 + \beta_3)} \leq \alpha'' t^2 + \beta''$$

as in the right-hand-side of (5.61), where

$$\beta'' \doteq \sqrt{\beta_2 \beta_3} \quad \text{and} \quad \alpha'' \doteq \frac{\alpha_2 \beta_3 + \alpha_3 \beta_2}{2\sqrt{\beta_2 \beta_3}}.$$

CHAPTER 5.

As for a lower bound, if we define $\alpha^2 \doteq \alpha_2\alpha_3$ and $\beta^2 \doteq \beta_2\beta_3$ then

$$(\alpha t^2 + \beta)^2 = (\alpha_2\alpha_3)t^4 + 2\sqrt{\alpha_2\alpha_3\beta_2\beta_3}t^2 + \beta_2\beta_3. \quad (5.62)$$

But from the AM-GM inequality

$$\frac{a+b}{2} \geq \sqrt{ab}, \quad (5.63)$$

it follows that the middle term in (5.62) is less than $\alpha_2\beta_3 + \alpha_3\beta_2$. And so

$$(\alpha' t^2 + \beta')^2 \leq (\alpha_2 t^2 + \beta_2)(\alpha_3 t^2 + \beta_3)$$

where

$$\alpha' \doteq \sqrt{\alpha_2\alpha_3} \quad \text{and} \quad \beta' \doteq \sqrt{\beta_2\beta_3}$$

give the constants so that the lower bound in (5.61) hold. Therefore, we have

$$\begin{aligned} \int_{-1}^1 \frac{\alpha_1 t^2 + \beta_1}{(\alpha_2 + \alpha')t^2 + (\beta_2 + \beta')} dt &\leq \int_0^\pi \frac{k_1(\theta)}{h_1(\theta)(h_1(\theta) + h_2(\theta))} d\theta, \\ \int_0^\pi \frac{k_1(\theta)}{h_1(\theta)(h_1(\theta) + h_2(\theta))} d\theta &\leq \int_{-1}^1 \frac{\alpha_1 t^2 + \beta_1}{(\alpha_2 + \alpha'')t^2 + (\beta_2 + \beta'')} dt. \end{aligned} \quad (5.64)$$

in the same way,

$$\begin{aligned} \int_{-1}^1 \frac{\alpha_4 t^2 + \beta_4}{(\alpha_3 + \alpha')t^2 + (\beta_3 + \beta')} dt &\leq \int_0^\pi \frac{k_2(\theta)}{h_2(\theta)(h_1(\theta) + h_2(\theta))} d\theta, \\ \int_0^\pi \frac{k_2(\theta)}{h_2(\theta)(h_1(\theta) + h_2(\theta))} d\theta &\leq \int_{-1}^1 \frac{\alpha_4 t^2 + \beta_4}{(\alpha_3 + \alpha'')t^2 + (\beta_3 + \beta'')} dt. \end{aligned} \quad (5.65)$$

CHAPTER 5.

Finally, we can bound M as

$$M_{lb1} = 4\pi \left(\frac{\alpha_1}{\alpha_2 + \alpha'} - \frac{(\alpha_1(\beta_2 + \beta') - \beta_1(\alpha_2 + \alpha')) \tan^{-1}\left(\sqrt{\frac{\alpha_2 + \alpha'}{\beta_2 + \beta'}}\right)}{(\alpha_2 + \alpha')\sqrt{(\alpha_2 + \alpha')(\beta_2 + \beta')}} \right. \\ \left. + \frac{\alpha_4}{\alpha_3 + \alpha'} - \frac{(\alpha_4(\beta_3 + \beta') - \beta_4(\alpha_3 + \alpha')) \tan^{-1}\left(\sqrt{\frac{\alpha_3 + \alpha'}{\beta_3 + \beta'}}\right)}{(\alpha_3 + \alpha')\sqrt{(\alpha_3 + \alpha')(\beta_3 + \beta')}} \right), \quad (5.66)$$

$$M_{ub1} = 4\pi \left(\frac{\alpha_1}{\alpha_2 + \alpha''} - \frac{(\alpha_1(\beta_2 + \beta'') - \beta_1(\alpha_2 + \alpha'')) \tan^{-1}\left(\sqrt{\frac{\alpha_2 + \alpha''}{\beta_2 + \beta''}}\right)}{(\alpha_2 + \alpha'')\sqrt{(\alpha_2 + \alpha'')(\beta_2 + \beta'')}} \right. \\ \left. + \frac{\alpha_4}{\alpha_3 + \alpha''} - \frac{(\alpha_4(\beta_3 + \beta'') - \beta_4(\alpha_3 + \alpha'')) \tan^{-1}\left(\sqrt{\frac{\alpha_3 + \alpha''}{\beta_3 + \beta''}}\right)}{(\alpha_3 + \alpha'')\sqrt{(\alpha_3 + \alpha'')(\beta_3 + \beta'')}} \right). \quad (5.67)$$

We now explore a second set of bounds on M for triaxial ellipsoids. Since the total mean curvature of uniaxial ellipsoids is known as exact expressions in both the prolate and oblate cases, by inscribing and circumscribing the tightest uniaxial ellipsoids around a given triaxial ellipsoid, we can obtain bounds on M of the form

$$M_{lb2} \leq M(\partial B) \leq M_{ub2} \quad (5.68)$$

The explicit formulas for the total mean curvature of uniaxial ellipsoids are given below. If $a = b = R$ and $c = \lambda R$ with $0 < \lambda < 1$, then [156]

$$M = 2\pi R \left[\lambda + \frac{\arccos \lambda}{\sqrt{1 - \lambda^2}} \right]. \quad (5.69)$$

When $\lambda > 1$,

$$M = 2\pi R \left[\lambda + \frac{\log(\lambda + \sqrt{\lambda^2 - 1})}{\sqrt{\lambda^2 - 1}} \right]. \quad (5.70)$$

To circumscribe a tightest uniaxial ellipsoid around a given triaxial ellipsoid with $a \leq b \leq c$, we stretch the triaxial ellipsoid in x -axis until the semi-axis length changes from a to

CHAPTER 5.

b , or we can stretch it in y -axis until the semi-axis length changes from b to c . The tightest circumscribed uniaxial ellipsoid can be chosen as the one with the smallest M and this M (denoted as M_{ub2} in this method) provides an upper bound of the M of the triaxial ellipsoid.

We can inscribe a uniaxial ellipsoid into the triaxial ellipsoid in the similar way by shrinking it in z -axis until the semi-axis length changes from c to b and the M of this uniaxial ellipsoids provides a lower bound of the M of the triaxial ellipsoid. However, we notice that an even tighter lower bound can be found by using the ‘‘Schwartz-symmetrized’’-version of the triaxial ellipsoid instead of a fully inscribed version. The ‘‘Schwartz-symmetrized’’-version of a convex body B , denoted as B^* can be generated by condensing B into circular cross sections along an axis through the center of B . Each line keeps its original area, and so $V(B) = V(B^*)$. By the fact of $M(^*) \leq M(B)$, we can obtain a lower bound of M and it is tighter than the one from the inscribed ellipsoid. For our ellipsoidal case, we can just let the semi-axis lengths of the ‘‘Schwartz-symmetrized’’ version of the triaxial ellipsoid either be $a^* = a, b^* = c^* = \sqrt{bc}$ or $a^* = b^* = \sqrt{ab} = c$ and the one with larger M is chosen as the lower bound of the M (denoted as M_{lb2}).

In practice, we choose the tightest bounds as our bounds of M , i.e.,

$$M_{lb} = \max\{M_{lb1}, M_{lb2}\}, \quad (5.71)$$

$$M_{ub} = \min\{M_{ub1}, M_{ub2}\}.$$

To better illustrate different bounds of M , we use 100 random trials, in which the semi-axis lengths and rotation angles of the ellipsoids are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. The results of M_{ub1} , M_{ub2} and M_{lb1} , M_{lb2} are

CHAPTER 5.

compared with the numerically calculated M (treated as the exact value of M) in Fig. 5.5.

To illustrate how the ratios among different semi-axis lengths affect the performance of the

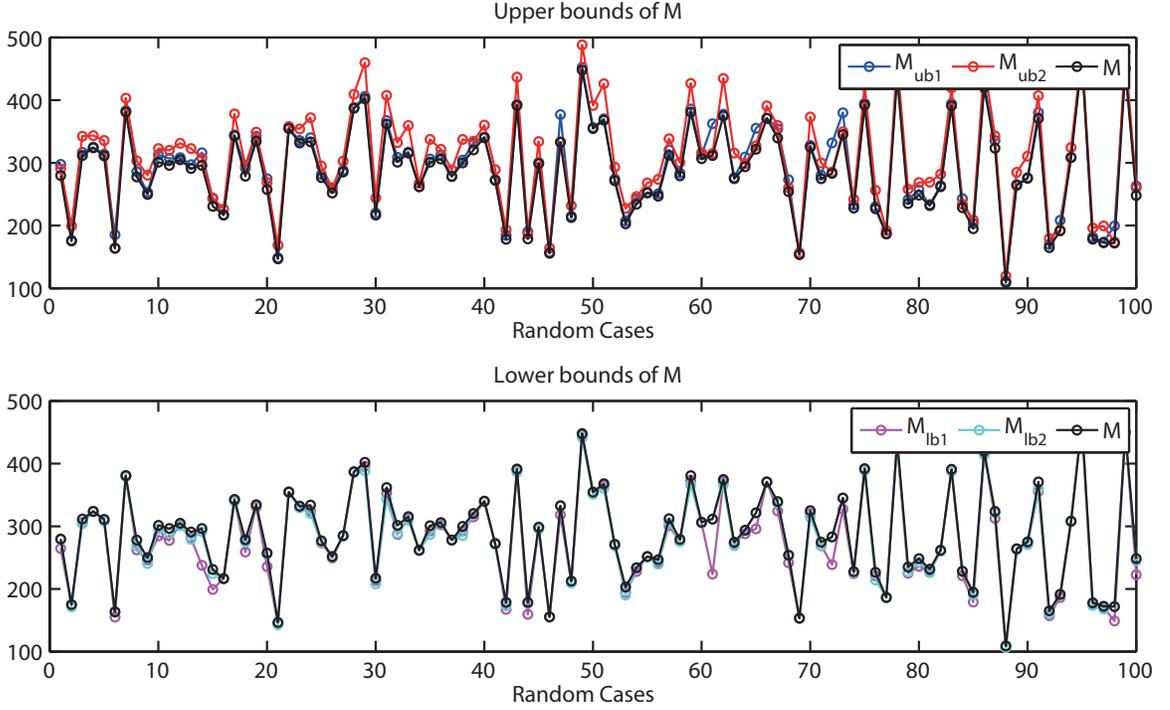


Figure 5.5: Upper and lower bounds, M_{ub1} , M_{ub2} , M_{lb1} and M_{lb2} , are compared with the exact values of M in 100 random trials, in which the semi-axis lengths and rotation angles of the ellipsoids are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively.

bounds of M , we also plot the relative errors of the bounds of M , i.e.,

$$e_{M_{ub}} = (M_{ub} - M)/M, \quad (5.72)$$

$$e_{M_{lb}} = (M_{lb} - M)/M,$$

On a 2D grid plot, with the axes of b/a and c/a . Here, b/a and $c/a \in [1, 5]$, represent the ratios between the semi-axis lengths b , a and c , a , respectively (see Fig. 5.6). In all the trials, the errors of the bounds are less than 5%. M_{lb1} and M_{ub1} always provide very tight

CHAPTER 5.

bounds. M_{lb2} and M_{ub2} perform better when the triaxial ellipsoid is close to a uniaxial one, and for the uniaxial case, the error becomes absolute zero.

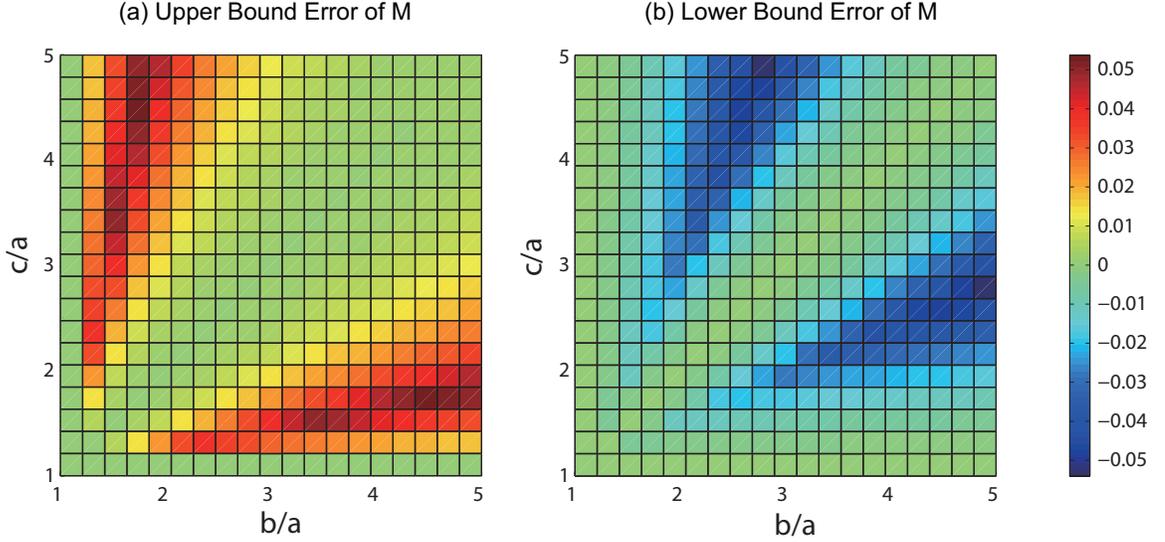


Figure 5.6: With different aspect ratios of semi-axis lengths, errors of M_{ub} and M_{lb} are shown on a 2D grid plot, with the axes of b/a and c/a , where $b/a, c/a \in [1, 5]$.

5.4.2.2 Bounds on the surface area of a triaxial ellipsoid

The surface area of a triaxial ellipsoid with $a \leq b \leq c$ can be written exactly as [157]

$$F(a, b, c) = 2\pi \left(a^2 + \frac{ba^2}{\sqrt{c^2 - a^2}} F(\varphi, w) + b\sqrt{c^2 - a^2} E(\varphi, w) \right) \quad (5.73)$$

where

$$w = \frac{c^2(b^2 - a^2)}{b^2(c^2 - a^2)} \text{ and } \varphi = \sin^{-1} \left(\sqrt{1 - \frac{a^2}{c^2}} \right).$$

Here $F(\varphi, w)$ and $E(\varphi, w)$ are respectively the incomplete elliptic integrals of the first and second kind. These integrals are built into packages such as `functions` in Matlab, and values can be looked up efficiently. Nevertheless, it can be useful to evaluate bounds in

CHAPTER 5.

terms of elementary functions.

$$F_{lb}(\partial B) \leq F(\partial B) \leq F_{ub}(\partial B). \quad (5.74)$$

Numerous papers provide lower and upper bounds on the surface area of ellipsoids [140, 151, 157, 158].

A convenient and very tighter upper bound of F can be found from the Cauchy-Schwartz inequality [154, 159].

$$F_{ub1} = \frac{4\pi}{\sqrt{3}} (a^2b^2 + b^2c^2 + c^2a^2)^{\frac{1}{2}}. \quad (5.75)$$

Since the explicit formulas for the surface area of uniaxial ellipsoids are also known as follows. If $a = b = R$ and $c = \lambda R$ with $0 < \lambda < 1$, then [156]

$$F = 2\pi R^2 \left[1 + \frac{\lambda^2}{\sqrt{1-\lambda^2}} \log \left(\frac{1 + \sqrt{1-\lambda^2}}{\lambda} \right) \right].$$

When $\lambda > 1$,

$$F = 2\pi R^2 \left[1 + \frac{\lambda^2 \arccos(1/\lambda)}{\sqrt{\lambda^2 - 1}} \right].$$

We can also circumscribe a tightest uniaxial ellipsoid around a given triaxial ellipsoid as we did for M to obtain an upper bound of F , denoted as F_{ub2} . Moreover, using the same ‘‘Schwartz-symmetrized’’ uniaxial ellipsoidal approximation, as what we did for M and the fact that $F(B^*) \leq F(B)$, we can obtain a tight lower bound of F , denoted as F_{lb} . To compare the results, we also use the same 100 random trials for F . The results of F_{ub1} , F_{ub2} and F_{lb} are compared with the exact value of F in (5.73) in Fig. 5.7.

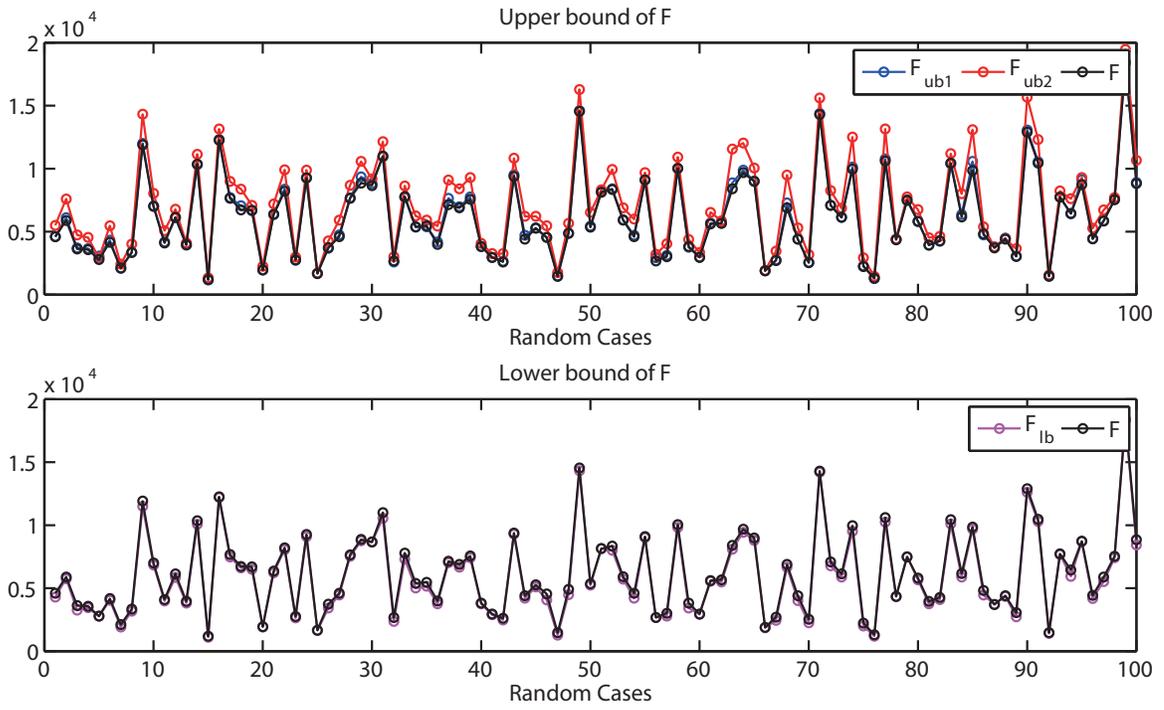


Figure 5.7: Upper and lower bounds, F_{ub1} , F_{ub2} and F_{lb} , are compared with the exact values of F in 100 random trials, in which the semi-axis lengths and rotation angles of the ellipsoids are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively.

CHAPTER 5.

Similarly, we also plot the relative errors of the bounds of F , i.e.,

$$e_{F_{ub}} = (F_{ub} - F)/F, \quad (5.76)$$

$$e_{F_{lb}} = (F_{lb} - F)/F,$$

on a 2D grid plot with the axes of b/a and c/a (see Fig. 5.8). Here,

$$F_{ub} = \max\{F_{ub1}, F_{ub2}\}. \quad (5.77)$$

Among all the trials, the errors of the bounds of F are always than 10%. The error becomes smaller when the triaxial ellipsoid is very close to a uniaxial one, and for the uniaxial case, the error becomes absolute zero.

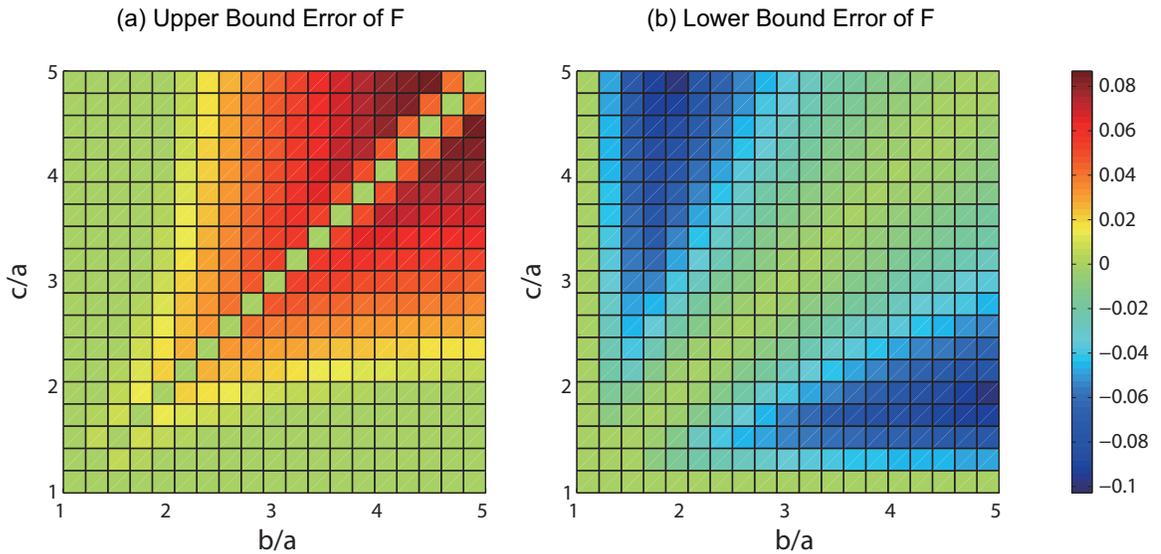


Figure 5.8: With different aspect ratios of semi-axis lengths, errors of F_{ub} and F_{lb} are shown on a 2D grid plot, with the axes of b/a and c/a , where $b/a, c/a \in [1, 5]$.

5.4.3 Numerical comparison of different bounds on the Minkowski sum volume of two ellipsoids

This section evaluates the tightness of the bounds derived in the previous sections. We begin with the planar case, and then address the spatial case.

5.4.3.1 The results for the planar case

For planar cases, we define bounds on the area enclosed by the Minkowski sum of ellipses as follows,

$$V_{BM} = \left(V(E_1)^{\frac{1}{2}} + V(E_2)^{\frac{1}{2}} \right)^{1/2}, \quad (5.78)$$

$$V_{lb} = \det(T) (V(B) + rL_{lb} + \pi r^2), \quad (5.79)$$

$$V_{ub} = \det(T) (V(B) + rL_{ub} + \pi r^2), \quad (5.80)$$

$$V_{exact} = \det(T) (V(B) + rL + \pi r^2), \quad (5.81)$$

where V_{BM} is a lower bound of the area from *Brunn-Minkowski* inequality (5.34)). V_{lb} and V_{ub} are the lower and upper bound by our Steiner's Formula-based approach. For planar cases, $V(B)$ becomes the area of the shrunk version of E_1 , i.e.,

$$V(B) = \pi a'_1 b'_1, \quad (5.82)$$

and $r = \min\{a_2, b_2\}$. L_{lb}, L_{ub} define in (5.45), (5.46) and L define in (5.42) are the upper, lower bounds and the exact formula for the perimeter of an ellipse, respectively.

CHAPTER 5.

Fig. 5.9 illustrates the Minkowski sums of two ellipses in two different cases. Fig. 5.10 shows the results of the relative errors for different bounds based on 100 trials, in which the semi-axis lengths and rotation angles of the ellipses are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. The relative error k_m for different bounds is defined as $k_m = (V_m - V_{true})/V_{true}$. In Fig. 5.10, we can see that V_{ub} and V_{lb} provide good upper and lower bounds with the relative errors less than 8%. We note that V_{BM} always provides a much looser lower bound than V_{lb} . To avoid unnecessary scaling effect on the figure we only plot V_{lb} , V_{ub} and V_{exact} in Fig. 5.10.

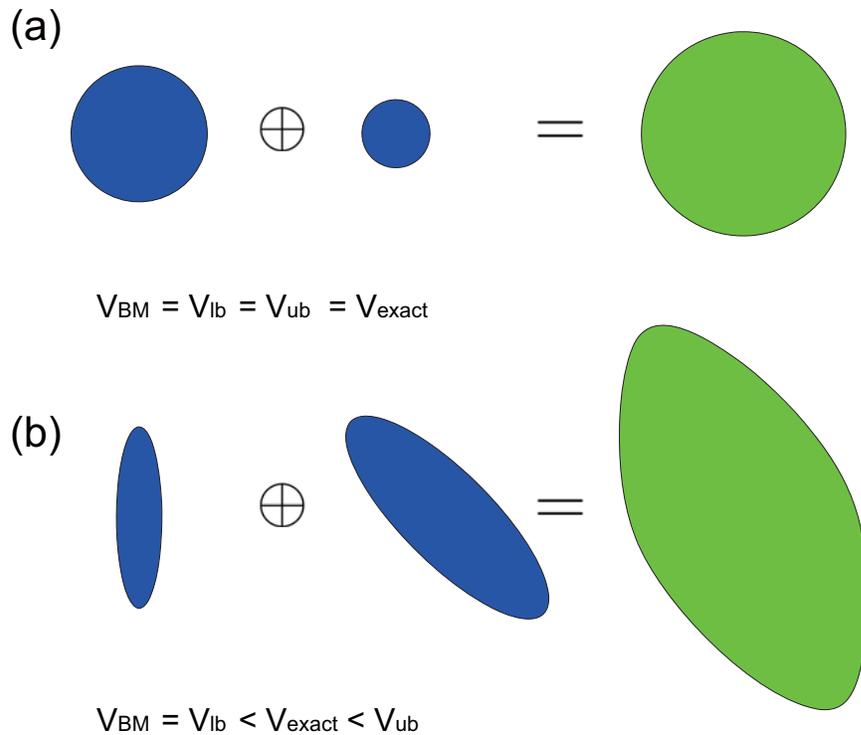


Figure 5.9: Comparisons of different bounds and exact values of Minkowski sums of two ellipses in two different cases. (a) Both ellipses are circles with radius 3 and 1.5, respectively. In this case, $V_{BM} = V_{lb} = V_{ub} = V_{exact} = \pi(3 + 1.5)^2 = 63.61$. (b) The ellipses have semi-axis lengths 1, 5 and 3, 6, and rotation angles 0 and $\pi/4$, respectively. $V_{BM} = 131.9 < V_{lb} = 165.2 < V_{exact} = 171.1 < V_{ub} = 179.1$.

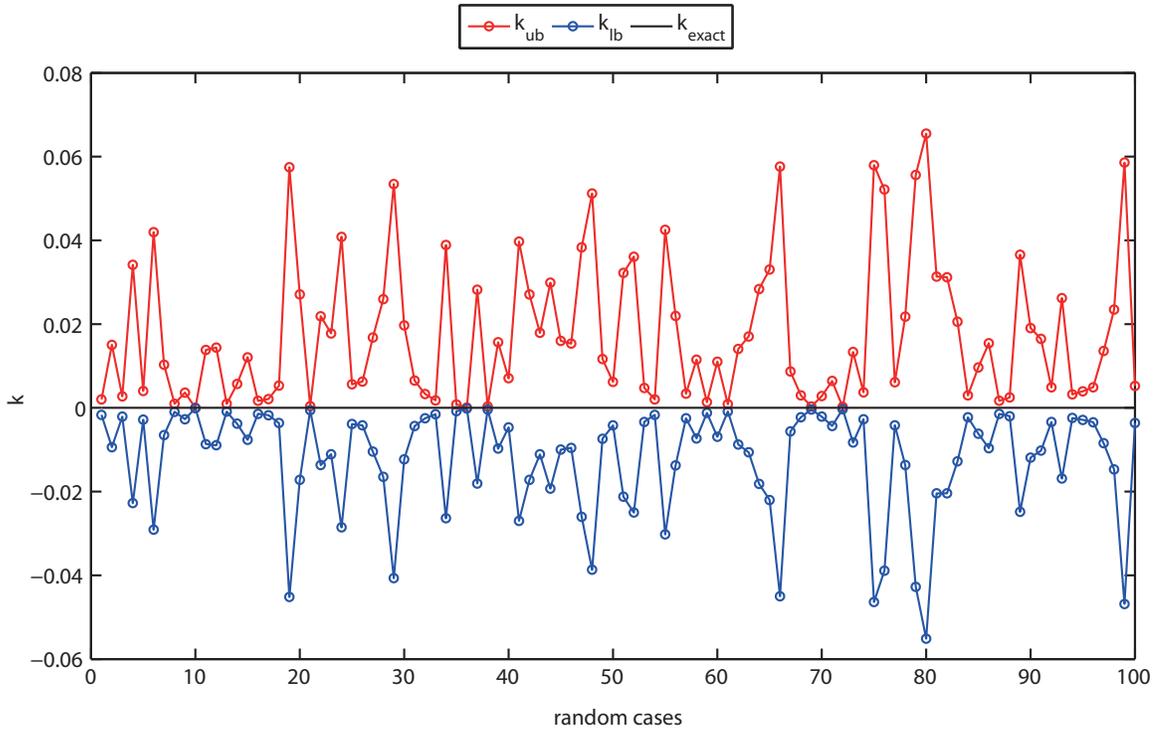


Figure 5.10: The upper and lower bounds, V_{ub} and V_{lb} , compared with the exact value of the Minkowski-sum volumes of two ellipses, V_{exact} , based on 100 trials, in which the semi-axis lengths and angles of the ellipses are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. The relative errors of the upper and lower bounds, k_{ub} and k_{lb} , are calculated and shown in the figure. Here, $k_{exact} = 0$.

5.4.3.2 The results for the spatial case

For three-dimensional cases, V_{BM} (based on the *Brunn-Minkowski* inequality (5.34)), V_{lb1} , V_{ub1} and V_{lb2} , V_{ub2} are define as follows,

$$V_{BM} = \left(V(E_1)^{\frac{1}{3}} + V(E_2)^{\frac{1}{3}} \right)^{1/3}, \quad (5.83)$$

$$V_{lb1} = \det(T) \left(V(B) + rF_{lb} + r^2M_{lb} + \frac{4\pi r^3}{3} \right), \quad (5.84)$$

$$V_{ub1} = \det(T) \left(V(B) + rF_{ub} + r^2M_{ub} + \frac{4\pi r^3}{3} \right), \quad (5.85)$$

$$V_{lb2} = \det(T) \left(V(B) + rF + r^2M_{lb} + \frac{4\pi r^3}{3} \right), \quad (5.86)$$

$$V_{ub2} = \det(T) \left(V(B) + rF + r^2M_{ub} + \frac{4\pi r^3}{3} \right), \quad (5.87)$$

$$V_{exact} = \det(T) \left(V(B) + rF + r^2M + \frac{4\pi r^3}{3} \right), \quad (5.88)$$

where $V(B)$ is the volume of the shrunk version of E_1 , with semi-axis lengths a'_1 , b'_1 and c'_1 , i.e.,

$$V(B) = \frac{4\pi}{3} a'_1 b'_1 c'_1, \quad (5.89)$$

and $r = \min\{a_2, b_2, c_2\}$. After calculating the offset surface, the “stretching” operation T can be found in (5.17). F is define in (5.73) as the exact surface area and the exact value of the total mean curvature M is calculated numerically based on (5.54). M_{lb} , M_{ub} and F_{lb} , F_{ub} are the best lower and upper bounds of the total mean curvature and the surface area, respectively. V_{lb1} and V_{ub1} use bounds for both M and F , while V_{lb2} and V_{ub2} use the exact value of F and the bounds of M .

Fig. 5.11 illustrates the Minkowski sums of two ellipsoids in three different cases. Fig. 5.12 shows the results of the relative errors for different bounds based on 100 trials, in

which the semi-axis lengths and rotation angles of the ellipsoids are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. With our best bounds of M and F , the relative errors of the Minkowski-sum volumes can be bounded by 6%. With the best bounds of M and the exact value of F calculated in closed form in (5.73), the relative errors can be further reduced and bounded by 2%.

5.4.4 Volume estimates on the Minkowski sum volume of two ellipsoids derived from the bounds

In addition to exact lower and upper bounds on the volume of Minkowski sums of ellipsoids based on our parametric description, we consider estimates of the volume. Two immediate estimates are obtained from taking the arithmetic and geometric means of our best lower and upper bounds V_{lb1} and V_{ub1} , induced by our best bounds of M and F :

$$V_{AM} = \frac{1}{2}(V_{lb1} + V_{ub1}), \quad (5.90)$$

$$V_{GM} = \sqrt{V_{lb1} \cdot V_{ub1}}. \quad (5.91)$$

From the AM-GM inequality (5.63), $V_{GM} \leq V_{AM}$.

Other estimates can be obtained by replacing the lower and upper bounds on F (Fig. 5.14), or in the planar case, L (Fig. 5.13) with their estimates.

For example, a modified Ramanujan approximation of the perimeter of an ellipse with

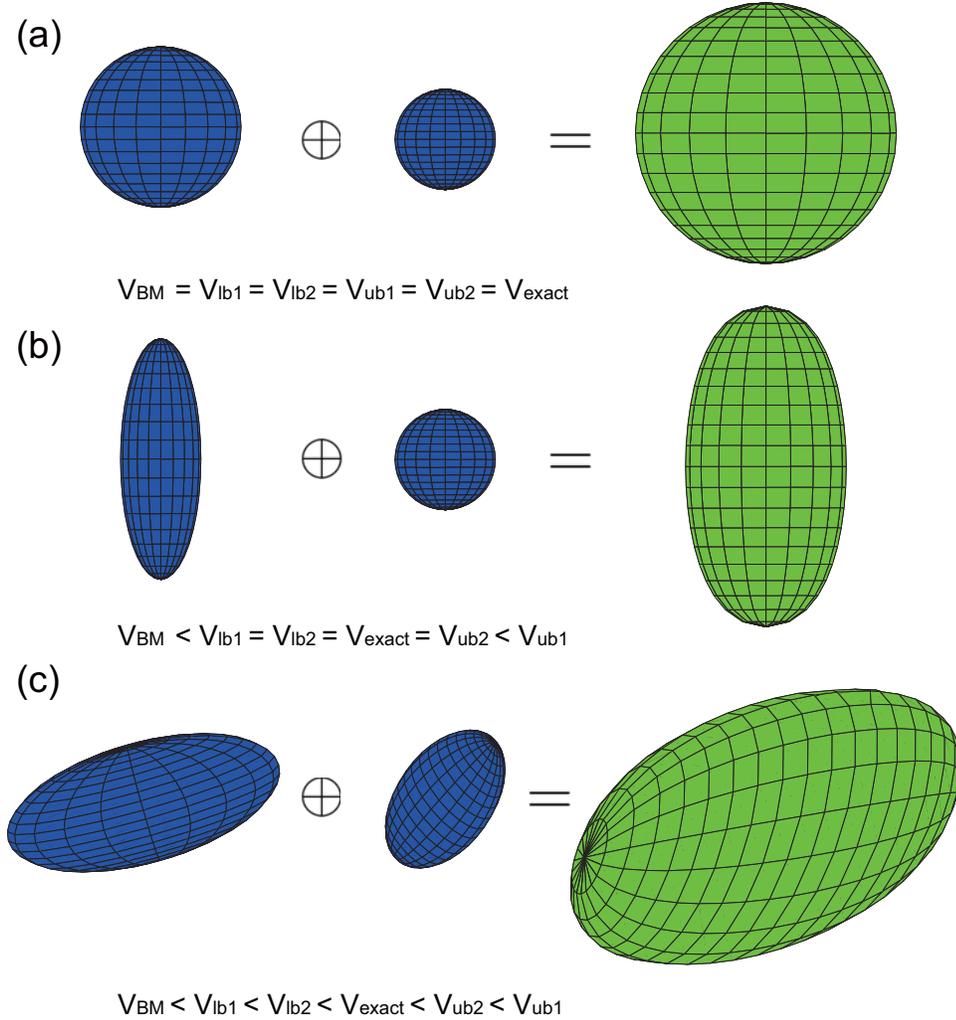


Figure 5.11: Comparisons of different bounds and exact values of Minkowski sums of two ellipsoids in three different cases. (a) Both ellipsoids are spheres with radius 4 and 2.5, respectively. In this case, $V_{BM} = V_{lb1} = V_{lb2} = V_{exact} = V_{ub2} = V_{ub1} = \frac{4\pi}{3}(4 + 2.5)^3 = 1.15 \times 10^3$. (b) One ellipsoid is a sphere with radius 2, the other one is axis-symmetric with semi-axis lengths 2, 2, 6. $V_{BM} = 448.1 < V_{lb1} = V_{lb2} = V_{exact} = V_{ub2} = 563.3 < V_{ub1} = 573.7$. (c) Both ellipsoids are triaxial with semi-axis lengths as 7, 3, 3 and 2, 3, 4, and ZXZ Euler angles as $-\pi/4, -\pi/8, \pi/4$ and $\pi/3, \pi/4, -\pi/6$, respectively. $V_{BM} = 1354.4 < V_{lb1} = 1678.8 < V_{lb2} = 1724.3 < V_{exact} = 1725.9 < V_{ub2} = 1729.2 < V_{ub1} = 1741.2$.

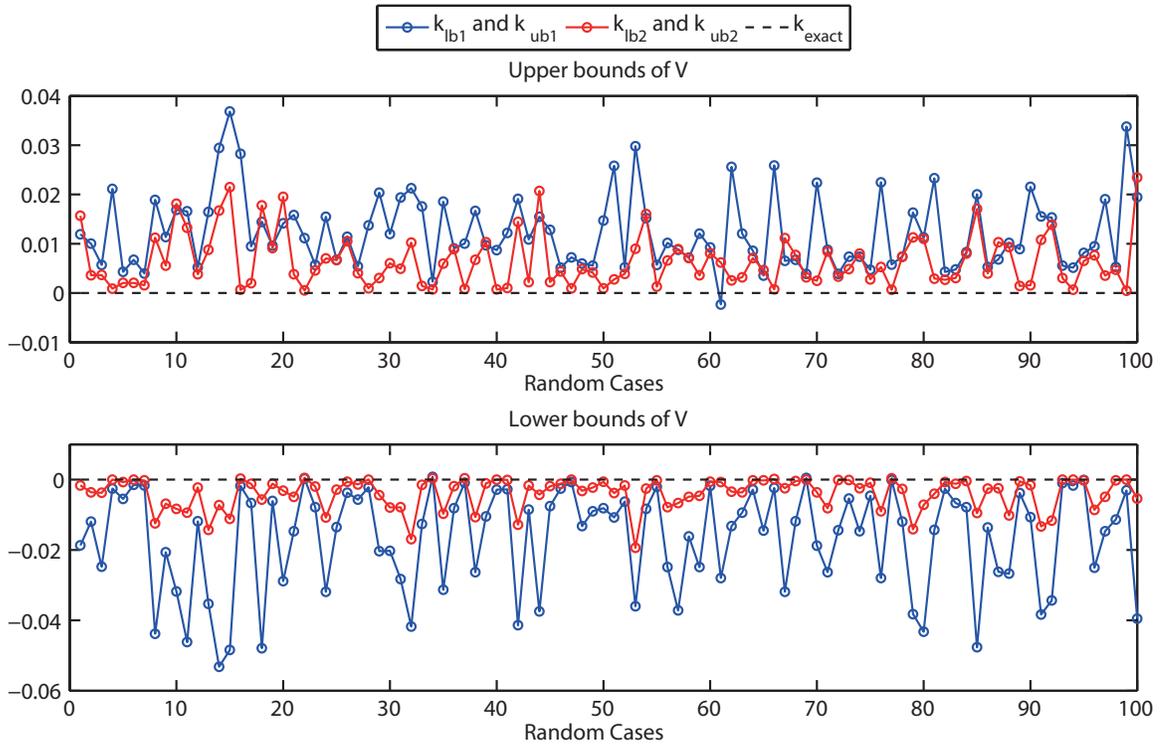


Figure 5.12: The upper and lower bounds, V_{ub1} , V_{ub2} , V_{lb1} and V_{lb2} , are compared with the exact value of the Minkowski-sum volumes, V_{exact} , based on 100 trials, in which the semi-axis lengths and angles of the ellipses are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively. The relative errors of these bounds, k_{ub1} , k_{ub2} , k_{lb1} and k_{lb2} , are calculated and shown in the figure. Here, $k_{exact} = 0$

CHAPTER 5.

$a \leq b$ is [160]

$$L_{est} = \pi(a+b) \left(1 + \frac{3(b-a)^2/(b+a)^2}{10 + \sqrt{4 - 3(b-a)^2/(b+a)^2}} \right). \quad (5.92)$$

An approximate formula for the area of a triaxial ellipsoid has been given recently (by

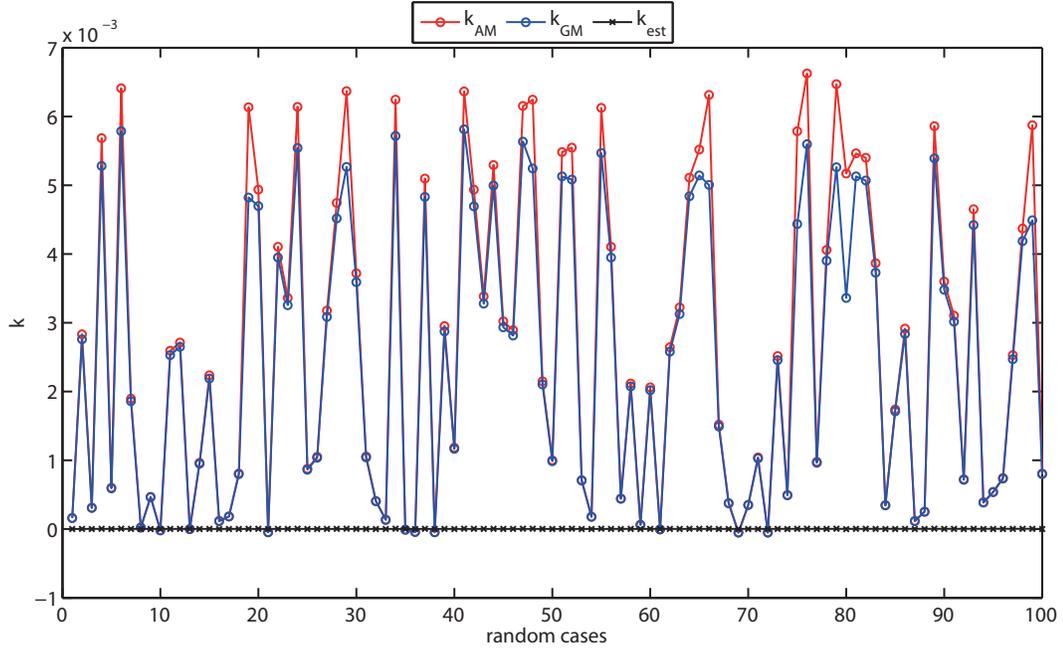


Figure 5.13: The relative errors of different area estimates, V_{AM} , V_{GM} and V_{est} , for $E_1 \oplus E_2$ are calculated based on 100 trials, in which the semi-axis lengths and angles of the ellipses are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively.

Thomsen and Cantrell independently) as [161]

$$F_{est} = 4\pi \left(\frac{a^p b^p + a^p c^p + b^p c^p}{3} \right)^{1/p} \quad \text{where } p = 1.6075. \quad (5.93)$$

With

$$M_{est} = \frac{1}{2}(M_{lb} + M_{ub}), \quad (5.94)$$

these can be used in Steiner's formula to obtain an estimate of the volume within an offset of an ellipse or ellipsoid.

CHAPTER 5.

The relative errors of the estimated volumes V_{AM} , V_{GM} and V_{est} of the 2D and 3D cases are compared in Figs. 5.13 and 5.14. In planar cases, the relative errors of V_{est} provides a very accurate estimation, i.e., $k_{est} \approx 0$. V_{AM} and V_{GM} , computed from our best lower and upper bounds V_{lb1} and V_{ub1} , are very good estimations as well, with the relative errors bounded by 0.7%. In spatial cases, the relative errors of V_{est} become larger. V_{AM} and V_{GM} in this case provide better estimations, with the relative errors always less than 1%.

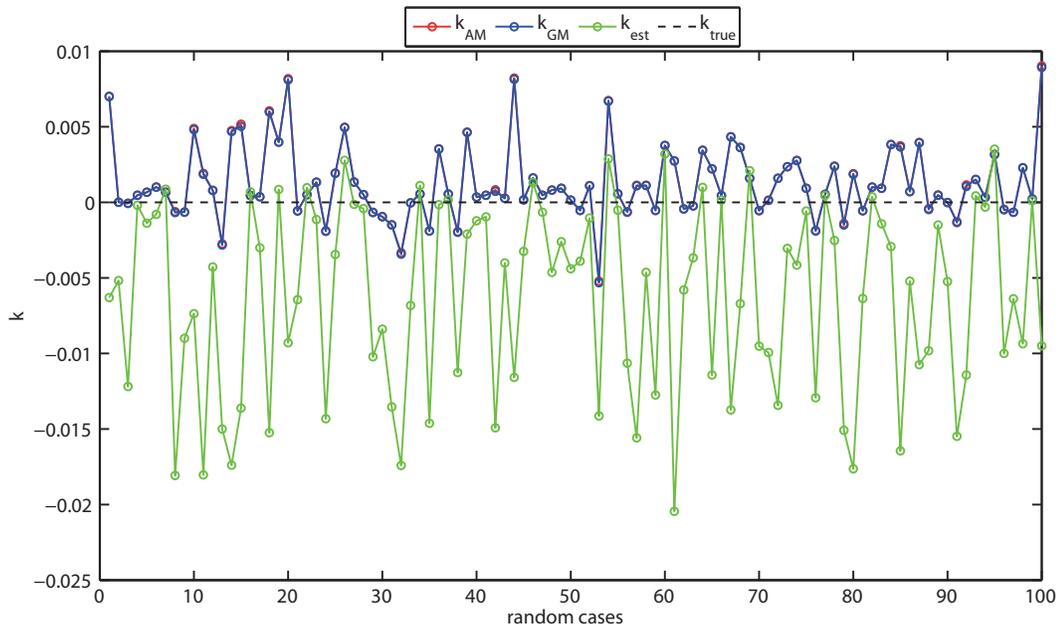


Figure 5.14: The relative errors of different volume estimates, V_{AM} , V_{GM} and V_{est} , for $E_1 \oplus E_2$ are calculated based on 100 trials, in which the semi-axis lengths and angles of the ellipsoids are randomly sampled from uniform distributions $U[10, 50]$ and $U[0, 2\pi]$, respectively.

5.5 A Kinematic Formula for Containment

In classical integral geometry, the Principal Kinematic Formula plays a central role. The formula expresses the average Euler characteristic of the intersections of rigid-bodies moving uniformly at random in terms of fundamental quantities of these bodies (volume, area, etc.). When the bodies are convex, the intersections are convex, and the Euler characteristic can be replaced by the set indicator function, $\iota(\cdot)$ which takes a value of 1 when the argument is nonempty, and 0 otherwise. The resulting formula works in \mathbb{R}^n and has been extended to general spaces of constant curvature. But we are concerned only with two- and three-dimensional Euclidean space, in which case the result is as follows.

5.5.1 The Formula for planar cases

Theorem 1 [162–164]): Given convex bodies C_0 and C_1 in \mathbb{R}^2 , then

$$\int_{SE(2)} \iota(C_0 \cap gC_1) dg = 2\pi[V(C_0) + V(C_1)] + L(\partial C_0) \cdot L(\partial C_1). \quad (5.95)$$

where $V(\cdot)$ is the area of the planar body and $L(\cdot)$ is its circumference. In \mathbb{R}^3

$$\int_{SE(3)} \iota(C_0 \cap gC_1) dg = 8\pi^2[V(C_0) + V(C_1)] + 2\pi[F(\partial C_0)M(\partial C_1) + F(\partial C_1)M(\partial C_0)] \quad (5.96)$$

where $F(\cdot)$ and $M(\cdot)$ are respectively the area and integral of mean curvature of the surface enclosing the body, and $V(\cdot)$ is the volume of the body. In these equations, $SE(n)$ denotes the $(n+1)n/2$ -dimensional Lie group of proper rigid-body motions in n -dimensional Eu-

CHAPTER 5.

clidean space, and dg denotes its (unnormalized) Haar measure. For the proof and pointers to the literature, see [162, 163, 165–169].

An alternative proof specifically for convex bodies was given in [103]. In that proof, the center of the moving body, C_1 , visits every point in the fixed body, C_0 , and rotates freely, each time contributing to the integral, and resulting in the $2\pi V(C_0)$ and $8\pi^2 V(C_0)$ terms. Then, the moving body is decomposed into concentric shells, and as each shell makes every possible point contact with the boundary ∂C_0 , intersections of the original bodies is also guaranteed. Adding up these contributions results in the above formulas.

This alternative proof is mentioned, because a new kind of kinematic formula can be derived in essentially the same way. In this new formula, we are concerned not with measuring the volume in $SE(n)$ corresponding to all possible intersections of bodies, but rather the integral of the volume in $SE(n)$ corresponding to all possible ways that C_1 can move while being fully contained in C_0 . To this end, let $b(gC_1 \subset C_0)$ take a value of 1 when $gC_1 \subset C_0$ and a value of zero otherwise, corresponding to the binary truth of the statement that the moving body is fully contained in the stationary one.

Theorem 2 Given convex bodies C_0 and C_1 in \mathbb{R}^n for $n = 2, 3$ such that C_0 can be written as the Minkowski sum $C'_0(R) \oplus RC_1$ for any $R \in SO(n)$ where $C'_0(R)$ is a convex body that depends on R , then

$$\int_{SE(2)} b((g \cdot C_1) \subset C_0) dg = 2\pi[V(C_0) + V(C_1)] - L(\partial C_0) \cdot L(\partial C_1) \quad (5.97)$$

CHAPTER 5.

and

$$\begin{aligned} & \int_{SE(3)} b((g \cdot C_1) \subset C_0) dg \\ &= 8\pi^2[V(C_0) - V(C_1)] - 2\pi F(\partial C_0)M(\partial C_1) + 2\pi F(\partial C_1)M(\partial C_0). \end{aligned} \quad (5.98)$$

For example, if C_0 and C_1 are circular disks in the plane with radii $r_0 > r_1$, then $L(\partial C_i) = 2\pi r_i$, $V(\partial C_i) = \pi r_i^2$ and the above planar formula (5.97) gives $2\pi^2(r_0 - r_1)^2$, which is 2π times the area of a disk of radius $r_0 - r_1$. Similarly, in the 3D case where $F(\partial C_i) = 4\pi r_i^2$, $M(\partial C_i) = 4\pi r_i$, $V(C_i) = \frac{4}{3}\pi r_i^3$, the above 3D formula (5.99) gives $8\pi^2$ times the volume of a sphere of radius $r_0 - r_1$.

We know of no other work that addresses this problem. The closest works are those of Zhou [170–173] that address when one body can be contained within another (but not the amount of motion allowed for a contained body). In some practical engineering contexts, this can be quite important [174–176].

Fig. 5.15 illustrates a planar example with the semi-axis lengths of E_1 and E_2 are 18, 15 and 2, 3, respectively. We numerically calculated the sum of all volumes of the Minkowski difference of two ellipse when one can move freely at all the orientations inside another without collision, i.e.,

$$V_{num} = \int_{SE(2)} V(E_1 \ominus (g \circ E_2)) dg. \quad (5.99)$$

In the example of Fig. 5.15, we compare V_{num} with V_{PK} , the volume calculated based on (5.97). The result shows that $V_{num} = V_{PK} = 3.799 \times 10^3$.

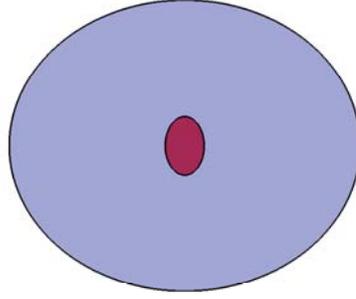


Figure 5.15: A planar example when one ellipse can move freely at all the orientations inside another without collision.

5.5.2 The Formula for spatial cases

The subject of translative kinematic formulas for general bodies that compute integrals of the form $\int_{\mathbb{R}^n} \iota(C_0 \cap tC_1) dt$ for convex bodies has been addressed extensively in [177–184]. Using our method, given ellipsoidal bodies E_0 and E_1 with

$$a_1 \leq b_1 \leq c_1 \leq a_0 \leq b_0 \leq c_0$$

we can compute a translative integral geometric formula for containment of the form

$$\int_{\mathbb{R}^n} b(tE_1 \subset E_0) dt = V(E_0 \ominus E_1).$$

The result of this formula is related to the formulas given in the previous section because

$$\int_{SE(3)} b((g \cdot E_1) \subset E_0) dg = \int_{SO(3)} V(E_0 \ominus (R \cdot E_1)) dR.$$

But using our closed-form translative kinematic formula for containment (which uses the Minkowski sum and difference results from earlier in the chapter), we can also compute what the volume of motion is in $SE(3)$ when the range of rotations is restricted. Fig. 5.16

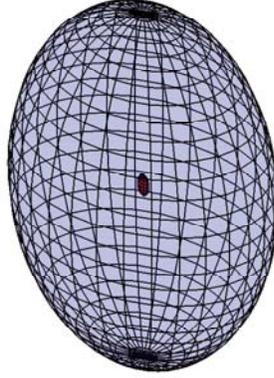


Figure 5.16: A spatial example when one ellipsoid can move freely at all the orientations inside another without collision.

illustrates a 3D example with the semi-axis lengths of E_1 and E_2 are 30, 40, 50 and 1, 2, 3, respectively. In this case, V_{num} is defined as

$$V_{num} = \int_{SE(3)} V(E_1 \ominus (g \circ E_2)) dg. \quad (5.100)$$

In this 3D example, we also compared V_{num} with V_{PK} (based on (5.98)) numerically. However, it is only feasible to discretize the 3D space of $SO(3)$ coarsely, and hence cross-validation is only approximate. With the resolution of integration in each degree of freedom defined by 50 sample points, we have $V_{num} = 1.57 \times 10^7$ and $V_{PK} = 1.64 \times 10^7$, which verify to within discretization error that these quantities are the same.

We note that if an ellipsoidal robot navigates within an ellipsoidal arena containing ellipsoidal obstacles, the methodology presented above can be used to compute the volume of collision-free motion in $SE(n)$ when certain conditions hold. In particular, if the obstacles are small enough and placed far enough away from the boundary of the arena such that it is never possible for the robot to simultaneously intersect two or more obstacles or

CHAPTER 5.

an obstacle and the boundary of the arena, then the volume of motions computed from our containment formula can be computed first and then the volume of motions computed from the Principal Kinematic Formula for the robot and each obstacle can be computed and subtracted. The result will be the volume of free motion. If the conditions mentioned above regarding the size and distribution of obstacles does not hold, then the result computed in this way will be an upper bound on the free motion. Moreover, using our bounds on the volume of Minkowski sums and differences, analogous quantities can be computed for the pure translative case (under less restrictive conditions on the size and location of obstacles).

5.6 Conclusion

In this chapter, we present an approach to parameterizing the exact boundaries of the Minkowski sum and difference of two ellipsoids. We also develop two other methods to quickly obtain the approximated Minkowski sum and difference based on the offset curve/surface approximations and special properties of the Gaussian distribution. In contrast to most existing methods, our approaches are completely analytical and have closed forms and therefore naturally provide improved efficiency and better robustness in many applications. With our exact parameterization, the volumes of the Minkowski sum and difference of two ellipsoids can be calculated efficiently and numerically, but in general, formulas for the volumes enclosed in these regions do not have exact closed-form expressions. Based on Steiner's Formula, we also develop a method to provide the upper and

CHAPTER 5.

lower bounds of the Minkowski-sum volumes. These bounds are within a few percent of the actual values over a wide range of aspect ratios. In the last part of the chapter, we also introduce a containment formula based on the Principal Kinematic Formula to compute the sum of all volumes of the Minkowski difference of two ellipsoids when one can move freely at all orientations inside another without collision.

Chapter 6

Path Planning Based on Characterization of Collision-Free Configuration-Space for Ellipsoidal Bodies, Obstacles, and Environment

In Chap. 5, we developed a method to parameterize the exact boundaries of the Minkowski sum and difference of two ellipsoids in closed-form. In the chapter, we apply this method in the context of robot motion planning. By using this method we can parameterize the collision-free configuration spaces (C-spaces) in robot path planning problem. We assume that the robot, obstacles, and environment can each be described as ellipsoids or finite unions of ellipsoids in the workspace. These objects can be quite general, including non-

CHAPTER 6.

convex bodies, and this approach represents an alternative to polyhedral representations of bodies. The approach builds on the core idea that the space of collision-free motions of one moving ellipsoid relative to one fixed ellipsoid can be characterized completely in closed form. This means that for ellipsoidal bodies, there is never any reason to sample and discard poses suspected of being in collision. The method is shown in both planar elliptical cases and more general ellipsoidal cases. In this chapter, parametric representations of the collision-free region C-space are given. Also, a “highway” roadmap system is constructed to connect the collision-free regions. Therefore collision checking can be eliminated not only for the vertices, but also for the edges, and then the problem is simplified into 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 some current sample-based path planning methods without *a priori* knowledge of free space. We can also use the knowledge of collision-free regions for single-rigid-bodies in path planning of articulated bodies. In this case, we construct a “safety function” to rapidly evaluate the collision of the articulated body by checking each single rigid link using the precomputed *a priori* knowledge of free space. This “safety function” approach alleviates the traditional time-consuming collision checking for high degrees of freedom (DOF) articulated bodies. It can be integrated with any existing sampling-based path planning methods and improve the computation speed, particularly in the narrow passage problem.

6.1 Introduction

This chapter presents a method that can be used in conjunction with existing path planning algorithms, particularly sampling-based methods and roadmaps based on cell decompositions. The basic approach is to describe 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 environment). Equipped with these tools, it becomes possible *a priori* “throw away” vast volumes within C-space that correspond to robot/obstacle or robot/environmental-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 space, the collision-free space can be directly decomposed into cells, 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 studied 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 free space and ‘one’ corresponding to obstacle regions) has been investigated [20, 185], as has

CHAPTER 6.

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, the method presented in this chapter 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 the individual ellipsoids. This is not to be confused with the problem of rapidly determining when two particular ellipsoidal bodies at given poses intersect or not [147–150], or the closely related problems of computing penetration depth for ellipsoids [186, 187] or polyhedra [188, 189]. Those methods require solution methods that are numerical in nature. In contrast, our formulation is *analytical*, resulting from methods of elementary differential geometry.

In the remainder of this chapter is structured as follows. Sec. 6.2 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 can be detected with a sweeping line and saved as collision-free intervals. Sec. 6.3 proposes an approach to build a highway roadmap system based on the midpoints of the collision-free intervals

CHAPTER 6.

along each sweep line. In Sec. 6.3.3, our path planner based on the characterization of collision-free space and the corresponding highway road system is applied in a planar example, in which all the robot, the obstacles and the environment are constructed as the finite unions of ellipsoids. By comparing the computational speed with some sampling-based path planning methods, the efficiency and the potential, especially for the narrow passage problem, is demonstrated. In Sec. 6.4, to solve the path planning problem for articulated bodies with high DOF, a hybrid approach integrated with existing sampling-based approaches with a “safety function” based on our *a priori* knowledge of collision-free C-spaces for single-rigid-bodies is proposed. Finally, Sec. 6.5 presents our conclusions.

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

In Chap. 5, the parametric equations of the collision-free boundaries of ellipsoid-ellipsoid interactions in two different types were derived.

In this Chapter, 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

CHAPTER 6.

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. 6.1 (a)). The boundaries of the collision-free space for the robot-obstacle interaction (blue) and those for the robot-environment interaction (red) are shown in Fig. 6.1 (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, \dots, E_k and g_2, g_3, \dots, g_k represent the rigid body motions between the first ellipse E_1 and the other ellipses E_2, E_3, \dots, 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, \dots, E_k be C_1, C_2, \dots, C_k , and then the collision-free space of the whole robot can be characterized as $C_1 \cap (g_2 \circ C_2) \cap (g_3 \circ C_3) \cap \dots \cap (g_k \circ C_k)$.

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 represented as

$$P_{CF} = P_{S_1} \cap P_{S_2} \cap P_{S_{n_s \cdot k}} - P_{O_1} \cup P_{O_2} \cup P_{O_{n_o \cdot k}}, \quad (6.1)$$

CHAPTER 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. 6.2.

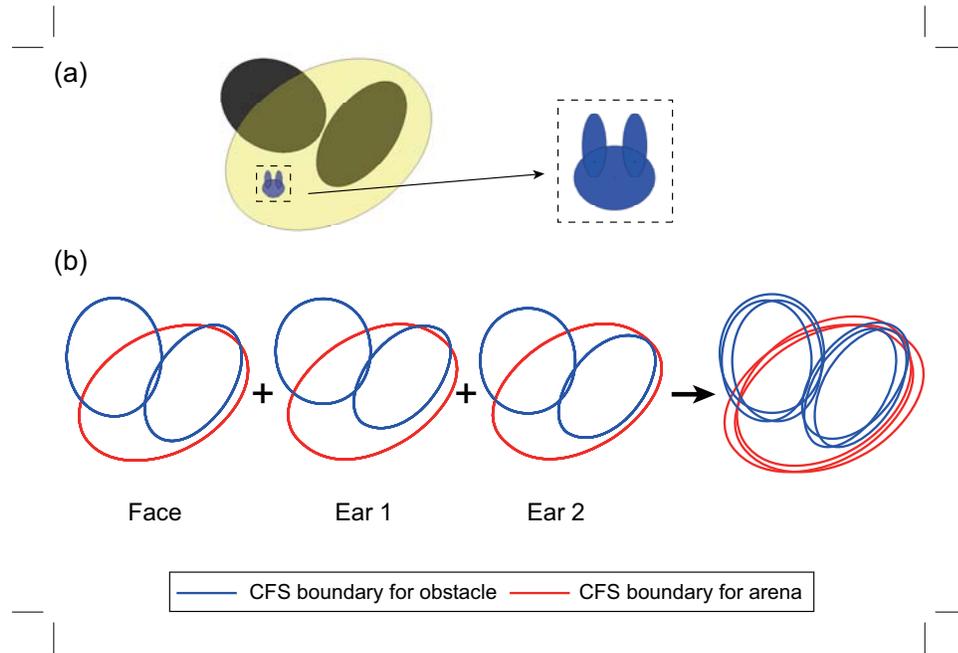


Figure 6.1: (a) The example of one rabbit-shaped robot translating inside an elliptical arena that contains elliptical obstacles. (b) The boundaries of the collision-free space (CFS) for the robot-obstacle interaction (blue curves) and those for the robot-environment interaction (red curves) for the rabbit face and two ears, respectively. The CFS is illustrated as the green-shaded region.

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 -coordinate, we are able to apply the detection scheme for the 2D elliptical case. A 3D example is shown in Fig. 6.3 and the resulting collision-free space using the detection

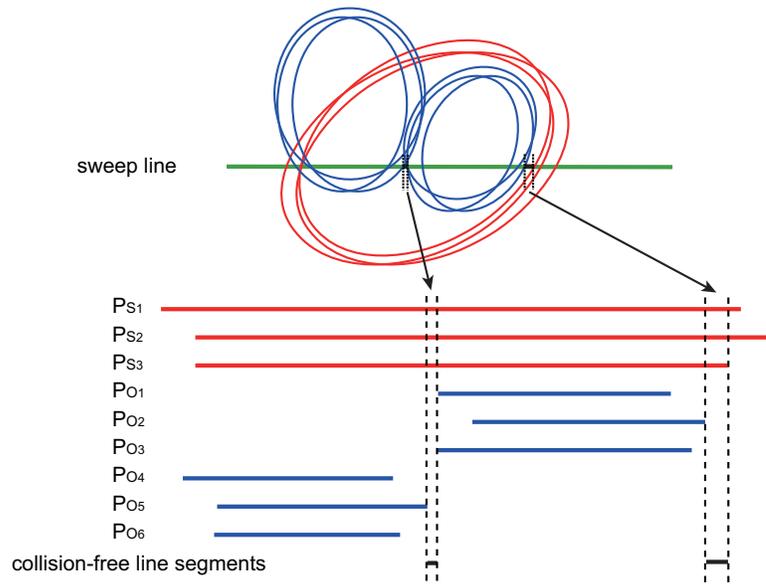


Figure 6.2: The detection scheme for the collision-free line segments on each sweep line using the logic in Eq. 6.1.

scheme is shown in Fig. 6.4.

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. 6.3), sampling resolution is not very critical, especially for smooth surfaces in our cases. Fig. 6.4 illustrates the collision-free space for the same example in Fig. 6.3. The collision-free space is constructed by stacking all the collision-free regions on each thin slice together along the z -axis.

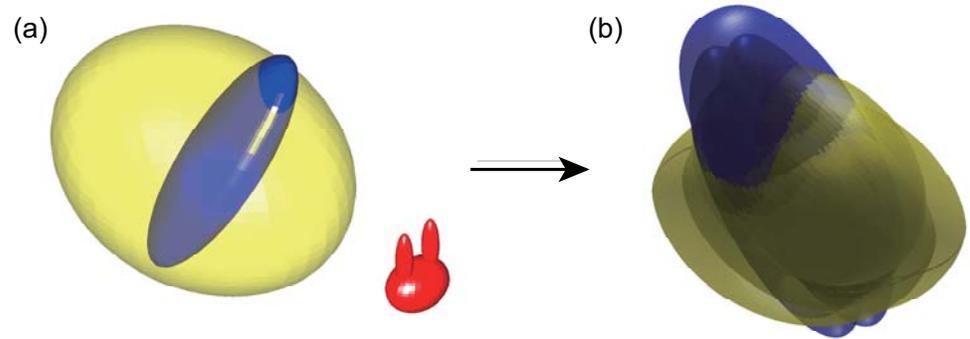


Figure 6.3: (a) The example of one rabbit-shaped robot translating inside an ellipsoidal arena that contains an ellipsoidal obstacle. (b) The overlaid boundaries of the collision-free space for the robot-obstacle interaction (blue-shaded surfaces) and those for the robot-environment interaction (yellow-shaded surfaces) for the rabbit face, ear 1 and ear 2.

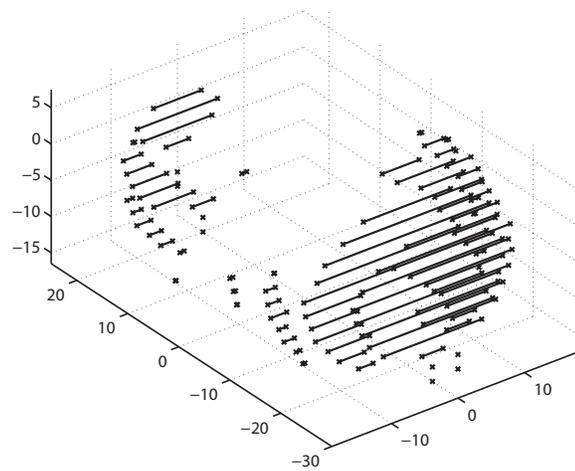


Figure 6.4: The collision-free space using the detection scheme for the example in Fig. 6.3.

6.3 Path Planning Approach 1: Highway Roadmap Planner

6.3.1 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 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.

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. 6.6 gives an example of how a planar roadmap

CHAPTER 6.

is connected on a x - y plane. When the robot has more DOF, a higher dimensional roadmap is needed. We also construct the rabbit-shaped robot with 5 DOF, 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. 6.5). In Fig. 6.7 (a), 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.

In the 3D case, we only consider that case when the rabbit is a rigid-body and only has translations. We still use the method illustrated in Fig. 6.6 to connect the road map for each x - y plane. After we capture a planar roadmap for each x - y plane, we need to connect the planar roadmaps along the z -axis. Here, we made an assumption that if two nodes on two different planes has a Euclidean distance smaller than L , the two nodes can be connected. Fig. 6.8 shows the 3D highway roadmap for the example in Fig. 6.3.

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 [190] to search for the shortest path.

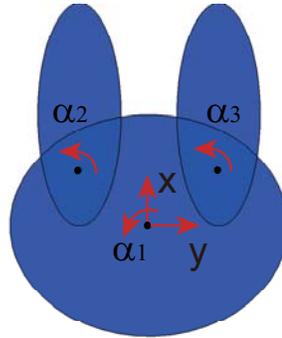


Figure 6.5: The rabbit-shaped robot with 5 degrees of freedom — the translation of the rabbit (in x and y directions) and rotations of the rabbit face and two ears (α_1 , α_2 and α_3), where $0 \leq \alpha_1 \leq 2\pi$, $0 \leq \alpha_2 \leq \pi/2$, and $-\pi/2 \leq \alpha_3 \leq 0$.

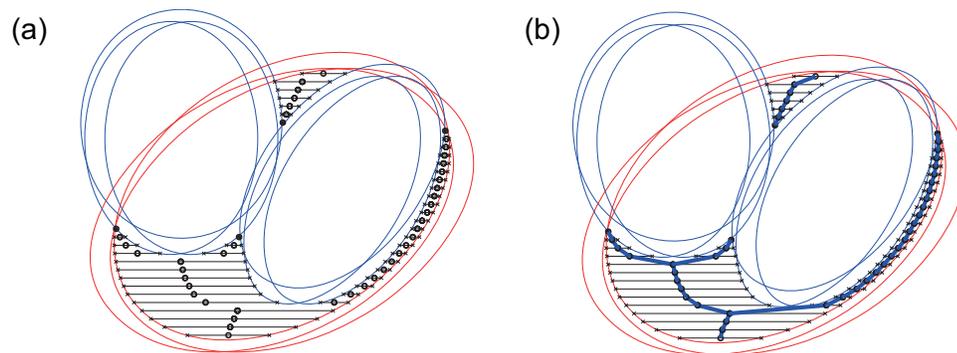


Figure 6.6: An example of generating the highway roadmap for the rabbit-shaped robot with 2 DOF (only translations in x and y directions). (a) The collision-free line segments and the midpoints on these line segments along each sweep line, (b) the adjacency graph connecting these midpoints, i.e., the highway roadmap, (shown as blue lines).

CHAPTER 6.

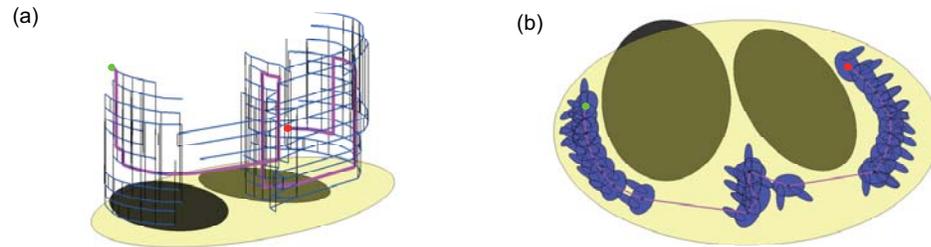


Figure 6.7: (a) An example of the highway roadmap for the robot with 5 DOF. Each layer of the roadmap representing a different combination of α_1 , α_2 and α_3 . The layers of the roadmap are interconnected at the nodes which are relatively close to one other (based on the Euclidean distance). A path is found using Dijkstra's algorithm given the start and goal (shown as magenta lines). (b) The steps of the motion of the robot overlaid onto the shortest path.

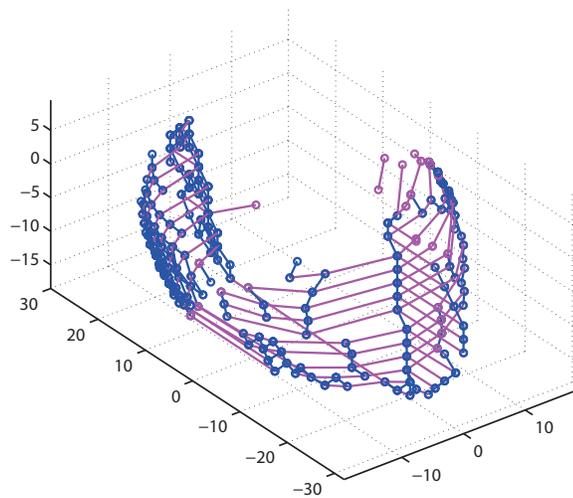


Figure 6.8: The 3D highway system for the example in Fig. 6.3.

6.3.2 Algorithm for removing redundant nodes on the Highway roadmap

In the 3D path planning problem, the number of nodes on the highway roadmap is substantially increased, especially when the robot has increased degrees of freedoms. Larger set of nodes will result in a larger connectivity mapping, and increase the time spent on the path searching. We notice that in the roadmap on each sliced plane, the existence of some nodes do not affect the connectivity of collision-free regions, but they do make the roadmap smoother. These nodes, called redundant nodes, can be removed to speed up the path searching. For each planar roadmap, we first separate it into different routes. Whenever the roadmap splits or converges, the new routes are generated (see Fig. 6.10 for an example of how to number the routes). For each single route, we do the redundant nodes removal separately. In this case, we can easily keep the intersecting nodes (where the roadmap splits or converges), and the conductivities near the intersections are therefore retained. Fig. 6.9 illustrates the procedure of removing the redundant nodes on each single route. Fig. 6.10 compares the planar roadmap before and after removing the redundant nodes. Using our algorithm, 196 redundant nodes are successfully removed out of 567 original nodes and the new connectivity mapping is constructed.

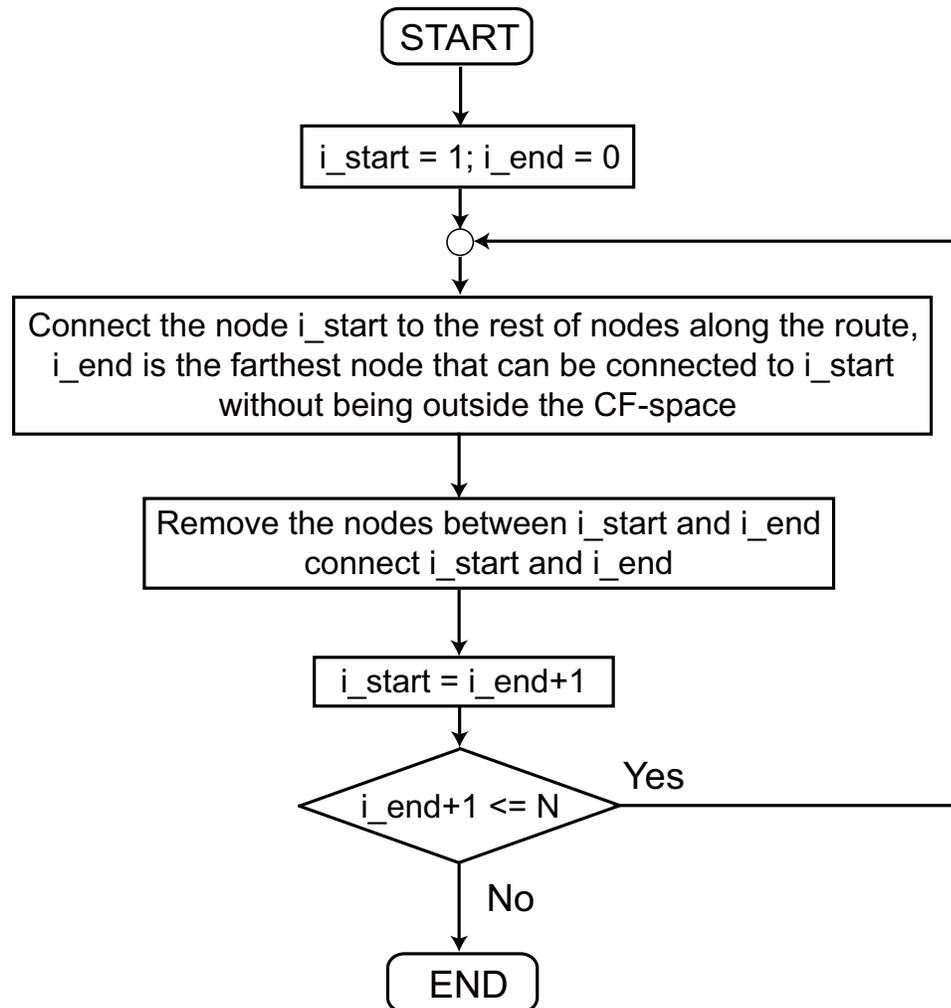


Figure 6.9: The procedure of removing the redundant nodes on each single route.

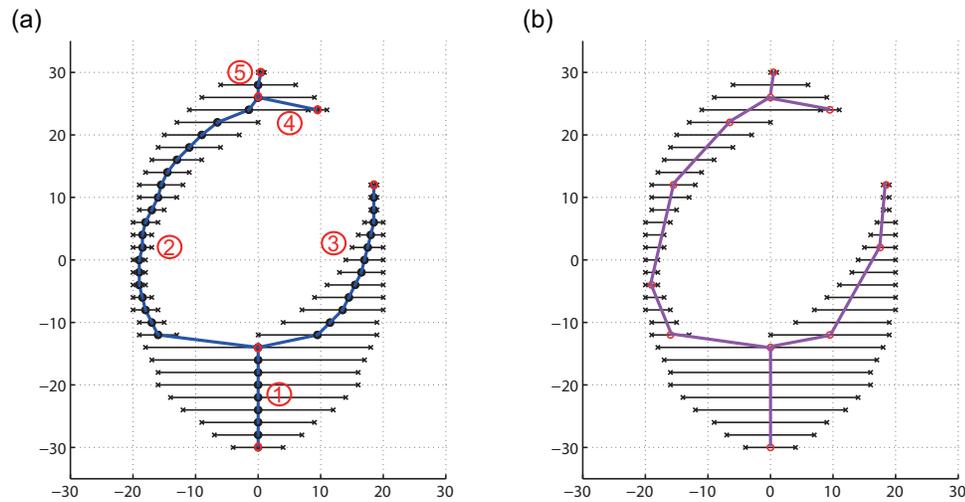


Figure 6.10: An example of a planar roadmap before and after removing the redundant nodes.

6.3.3 Path planning examples

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.

6.3.3.1 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. 6.7 (b) and 6.11). We compare our approach with two popular sampling-based path planning algorithms, a probabilistic roadmap [191] and a standard rapidly-exploring random tree (RRT) [53]. 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. 6.11). The codes are all written using MATLAB 2012b and run on a Lenovo Thinkpad T400 with Intel(R) Core(TM)2 Duo CPU P8700 Processor and 8 GB RAM.

In the experiments, 5 trials are generated using both the RRT and the PRM with the computational speeds shown in Tabs. 6.1 and 6.2. 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. 6.7), 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. However, as the free motion space of the robot significantly shrunk (as shown in Fig. 6.11), with our approach, it only takes 37.3 seconds to compute a feasible path, compared to an average speed of 1049.3 seconds using the RRT and 1346.4 seconds using the PRM.

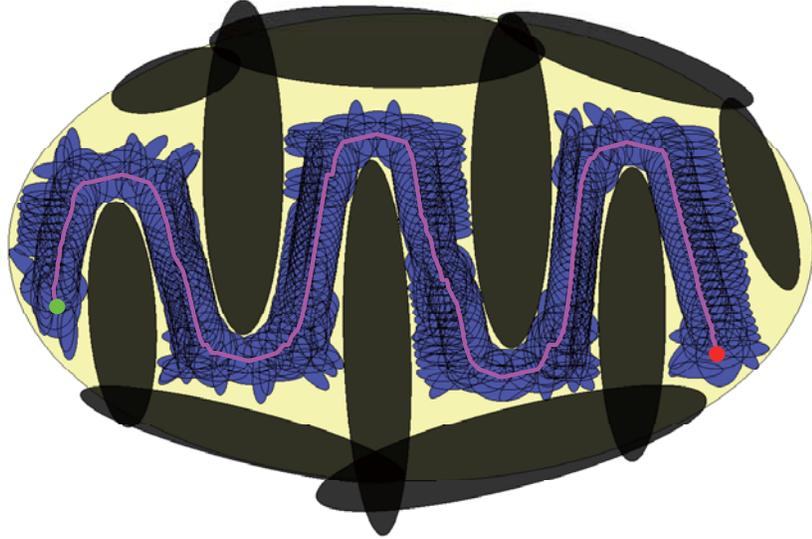


Figure 6.11: A planar example with the narrow passage problem. We note that the rabbit's two ears cannot fold back inward to the face since the rabbit model has the range of motion on the ears, i.e., $0 \leq \alpha_2 \leq \pi/2$, and $-\pi/2 \leq \alpha_3 \leq 0$.

Table 6.1: 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. 6.7 and 6.11, respectively.

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)	738	517	1842	884	1264	1049	37

CHAPTER 6.

Table 6.2: 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. 6.7 and 6.11, respectively. The numbers of nodes used in each trial to find a feasible path by the PRM are also shown in the table.

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
# of nodes	200	160	180	200	120	200	
Ex.2 time (s)	1138	1117	2243	1257	977	1346	37
# of nodes	3310	2660	4190	2100	1870	2740	

6.3.3.2 Spatial examples

In this section, a 3D example with the narrow passage problem is given. In this example, the robot, obstacle(s) and environment are constructed by unions of ellipsoids (see Fig. 6.12). The rabbit-shaped robot has fixed face and ear angles, i.e., only has translations. We also compare our approach with both the PRM and the RRT in finding a feasible path. With our approach, it still only takes 7.043 seconds to generate the highway roadmap and 0.032 seconds to find a feasible path. However, 212.7 seconds on average (based on 5 trials) are taken for the PRT, and 132.2 seconds for the PRM.

We also calculated the minimum distance between the robot and the whole environment (including the arena and obstacle(s)) at each step. This minimum distance is defined with a sign. The positive sign means the robot is not in collision and the distance is the minimum distance to the whole environment, while the negative sign means the robot is penetrating

the environment, and the distance represents the maximum penetration depth. See Fig. 6.13 for the minimum distance plots along the feasible paths between the starting points and goals.

The use of our new method is not an either-or proposition relative to sampling-based approaches. One can imagine how the full characterization of the free space can be used in conjunction with sampling-based approaches such as the RRT or the PRM rather than the highway approach that we have taken. Our purpose in providing this comparison is to illustrate that for worlds constructed from elliptical models (rather than polyhedra), the free-space can be characterized efficiently, and this has concrete computational implications. It demonstrates the great advantage of obtaining the information of collision-free C-space in advance and the improvement in computational speed in path planning, especially for narrow passage problems.

6.4 Path Planning Approach 2: Sampling-Based Approach with Safety Function for Articulated Bodies

While the highway roadmap works well with motion planning for single-rigid-bodies, especially for the narrow passage problem, it still suffers from the “curse of dimensionality”. In this section, for articulated bodies with high DOF, we propose a hybrid approach

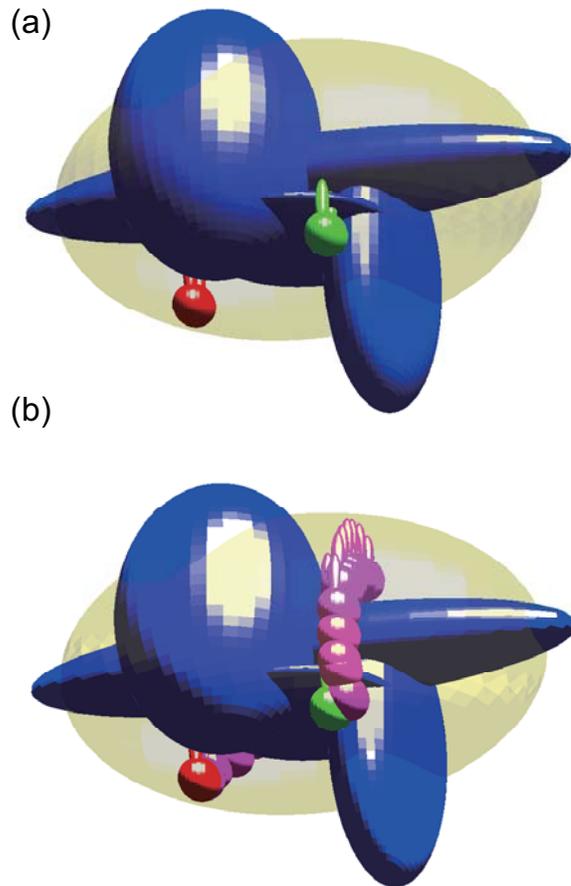


Figure 6.12: A 3D example is shown with the narrow passage problem. The red and green rabbits represent the starting point and the goal, and each magenta rabbit represents a step along the feasible path connecting the starting point and the goal. The path is generated by the highway roadmap approach.

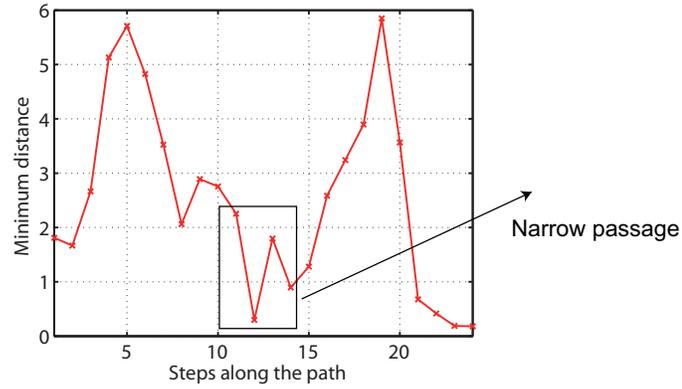


Figure 6.13: The minimum distance along the paths. The black square box highlights where the minimum distance significantly decreases and the narrow passage problem occurs.

integrated with some existing sampling-based approach with a safety function based on our *a priori* knowledge of collision-free C-spaces for single-rigid-bodies.

More specifically, we can save the collision-free C-space of a single-rigid-body as a 3D (for planar cases in $SE(2)$), or 6D (for spatial cases in $SE(3)$) “safety function” database that has values 1 when the configuration is out of the C-space obstacles and values 0 when inside of the the C-space obstacles. For an articulated body, the safety function can also be constructed. First, let $g_0 \in SE(3)$ denote the pose of the base body on the robot of an articulated structure. If the articulated structure is constructed from n bodies, each will have its “safety function” $s_i(g_i)$ where g_i is the pose of body i for $i = 0, \dots, n - 1$. If the internal DOF of the body are denoted as \mathbf{q} , the forward kinematic function from g_0 to g_i gives $g_0^{-1}g_i = f_i(\mathbf{q})$. That is, the frame of body i relative to the base body 0 is computed from the forward kinematics $f_i(\mathbf{q})$. Then the safety function for the whole articulated body will be $S(g_0; \mathbf{q}) = \prod_{i=0}^{n-1} s_i(g_0 \circ f_i(\mathbf{q}))$. By definition if the value of $S(g_0; \mathbf{q})$ equals 1, $(g_0; \mathbf{q})$ will be a collision-free configuration. When $S(g_0; \mathbf{q})$ equals 0, the articulated structure is in

CHAPTER 6.

collision.

Collision checking in the path planning could be computational expensive, especially for the robot with high DOF. By using the approach, no collision checking is needed and the complexity grows linearly with the number of articulated bodies increased. We note that it is especially useful when all the articulated bodies are identical, which means the collision-free C-space only need to be precomputed once for one single-rigid-body. This approach can replace the traditional collision checking procedure in the path planning and we can augment sampling methods such as the RRT or the PRM by providing a tool for rapid evaluation of feasibility of samples.

Fig. 6.14 (a) shows a planar example with a 5-link articulated robot arm. Each link is represented by an ellipse. The robot has 7 DOF (x and y : the translation of center of Ellipse 1 and θ_1 - θ_5 the rotations of Ellipse 1-5). To demonstrate our approach, a 2D maze example with 40 obstacles is constructed. A standard PRM is applied for both our safety function based approach and the traditional collision checking approach. The 2D maze with the starting point and the goal is shown in Fig. 6.15 (a). The roadmap with 3000 nodes is illustrated in Fig. 6.15 (b) and the feasible path connecting the starting point and goal is shown in Fig. 6.15 (c). For this robot arm, since all the links have the identical shape, we only need to construct the safety function database once (see an example of the safety function database with one fixed orientation in Fig. 6.16). The safety function for the whole robot arm $S = \prod_{i=0}^4 s_i$ was implemented in the PRM to evaluate the feasibility of samples. In this 2D maze example, 3000 nodes were generated for the PRM. By using our safety

CHAPTER 6.

function based approach, 434.8 seconds are taken to construct the roadmap, including the time spent on precomputed the safety function database. However, by using the traditional collision checking to construct the roadmap, 3056.9 seconds are used.

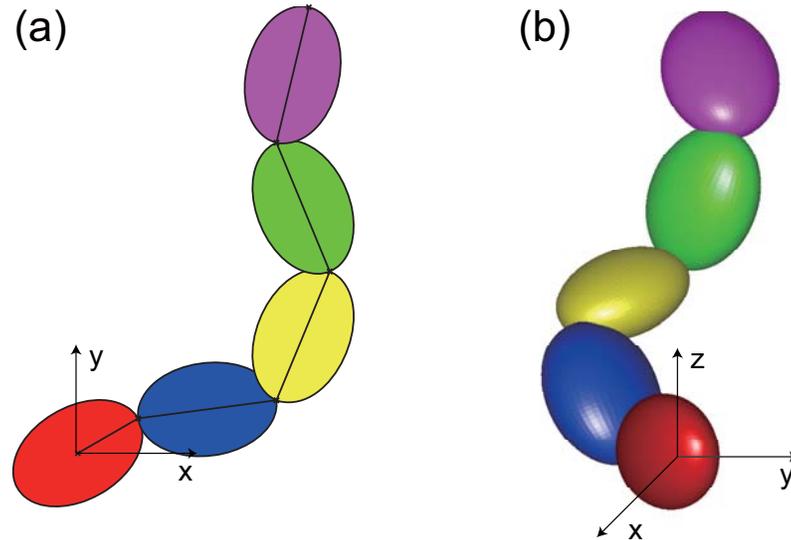


Figure 6.14: (a) The 5-link robot arm in the planar case with 7 DOF; (a) The 5-link robot arm in the spatial case with 18 DOF.

We also apply this approach to a spatial example with a 5-link articulated robot with each link represented by an ellipsoid (see Fig. 6.14 (b)). The robot has 18 DOF (3 DOF in the translations of the base ellipsoid, and 3 DOF in the rotations for each of the 5 ellipsoids). To demonstrate our approach, a 3D ellipsoidal environment clustered with ellipsoidal obstacles is used. To find the feasible path, the link robot has to path a narrow passage. We also compare our approach collaborated with the PRM to the PRM with traditional collision checking. For both approaches, 2000 nodes are sampled to construct the roadmap (see Fig. 6.17). In our approach, 456.7 seconds were used to construct the collision-free C-space for single ellipsoidal body with 6 DOF, and by using this precomputed collision-free C-space,

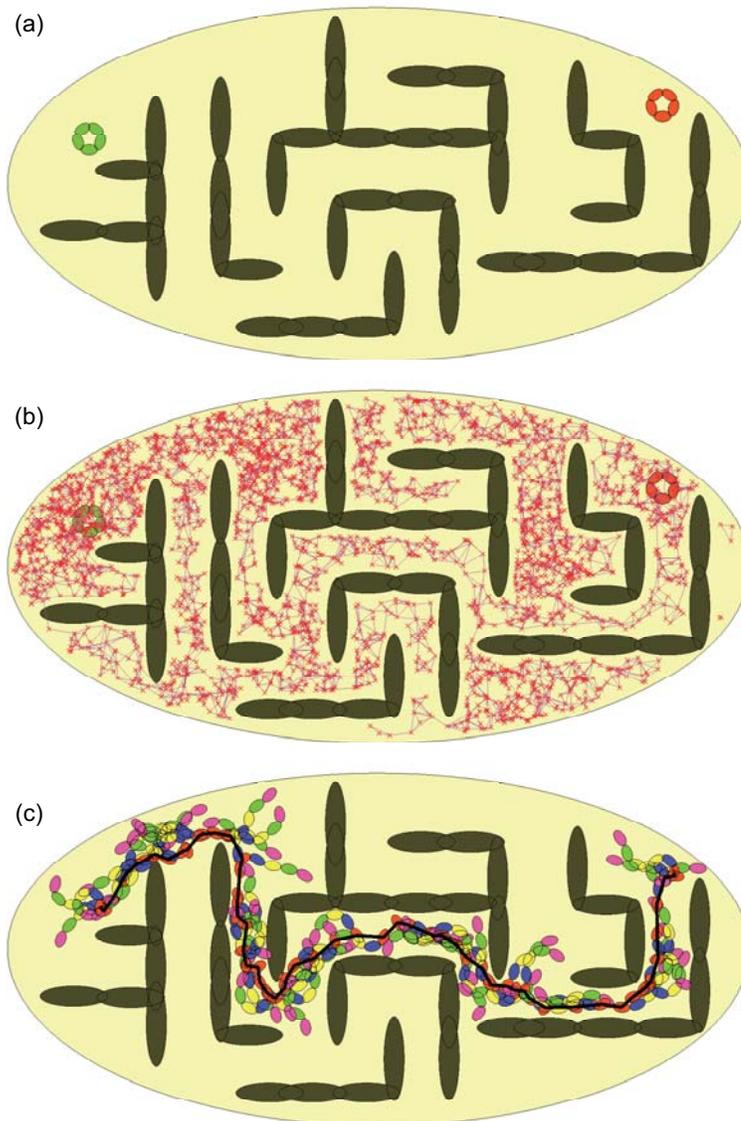


Figure 6.15: (a) The 2D maze environment; (b) The roadmap constructed for the 2D maze environment by using our safety function-based hybrid approach; (c) A feasible path found for the 2D maze environment.

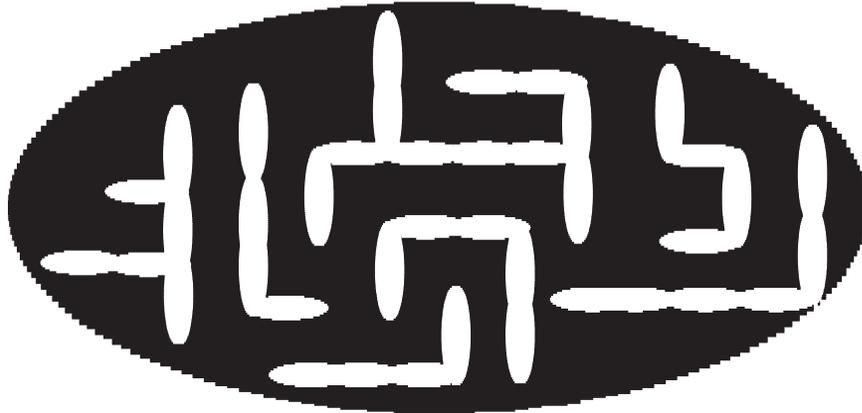


Figure 6.16: An example of one “slice” of the safety function database with $\theta = 0$ for a single elliptical link in the 5-link planar articulated robot. The safety function database is 3D for planar cases with each slice representing a collision-free C-space with a fixed rotation. The black shaded regions represent the collision-free C-space while the white-shaded regions represent the C-space obstacles.

240 seconds were used to construct the roadmap and find a feasible path (see the pose of the link robot in separate steps in Fig. 6.18). However, by using the PRM with traditional collision checking, 1074 seconds were used to construct the roadmap and find a feasible path.

6.5 Conclusion

In this Chapter, 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

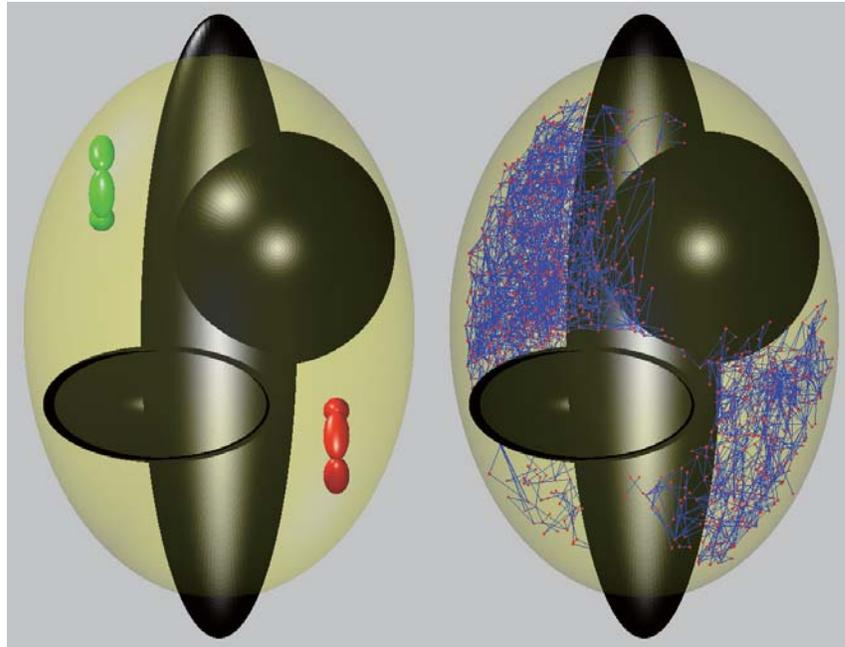


Figure 6.17: (a) A spatial path planning example is formed by a 5-link articulated robot roaming in an ellipsoidal environment clustered with ellipsoidal obstacles. The initial state (green) and the final state (red) of this 5-link articulated robot are defined and the robot has to pass a narrow passage to connect the initial state and the final state. (b) A roadmap of this path planning example is constructed by the PRM using 2000 nodes. In the roadmap, we can see that a narrow passage (with less nodes and connected edges) arises near the gap of the two small ellipsoidal obstacles.

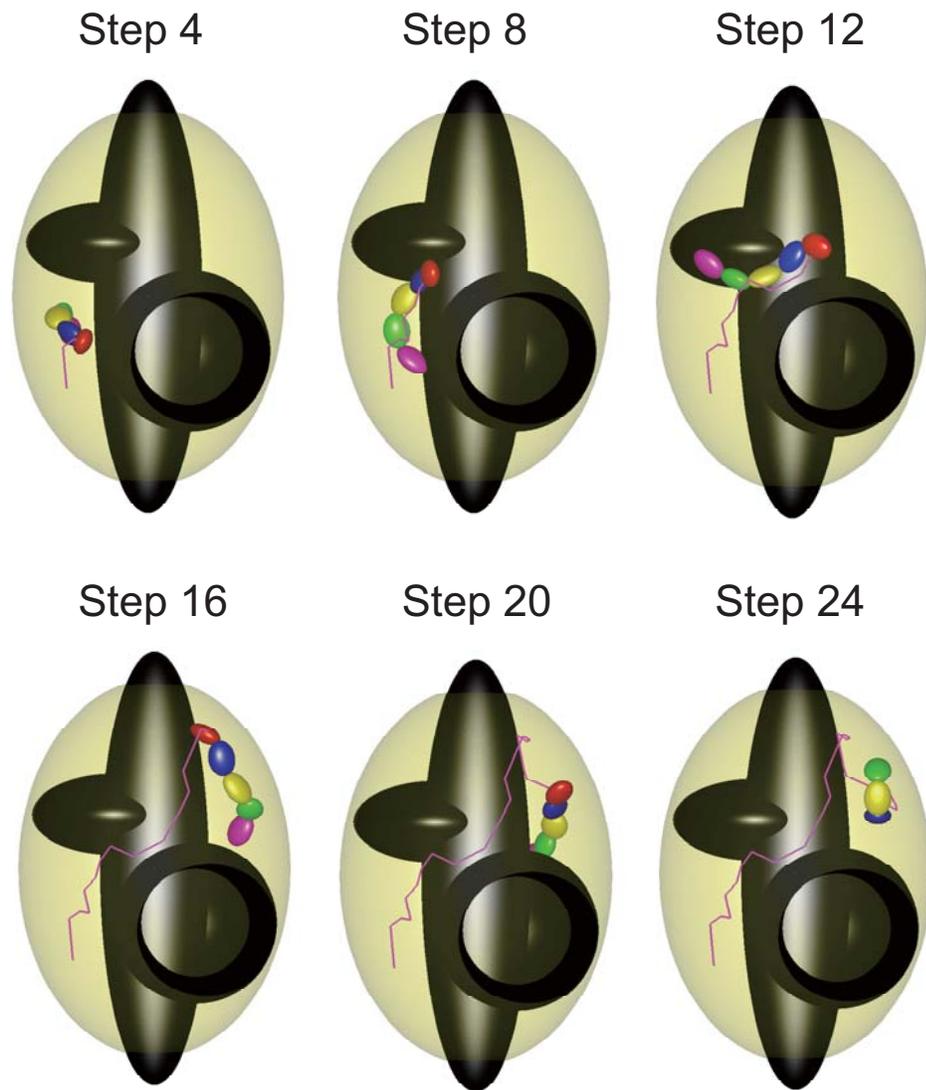


Figure 6.18: Using the PRM combined with our safety function-based detection scheme, a feasible path is found. Some of the intermediate steps are illustrated.

CHAPTER 6.

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. For articulated bodies, a hybrid approach integrated with existing sampling-based approaches with a “safety function” based on our *a priori* knowledge of collision-free regions for single-rigid-bodies is proposed. For both scenarios, we have demonstrated the efficacy and the potential of our method with both planar and spatial examples, especially for the narrow passage problem.

Chapter 7

Conclusion

This dissertation addresses geometric motion planning approaches in the contexts of robotics and biological crystallography. Useful geometrical models are incorporated into the characterization of collision-free configuration spaces for the moving objects. Applying the approach of “knowing where to look” for path planning in C-spaces helps improve the computational speed with the narrow passage problem. An almost uniform sampling approach for the rotation and motion spaces is also proposed in this dissertation to further facilitate the path planning efficiency. The work in this dissertation can be summarized in three areas:

- Macromolecular crystallography has been the traditional workhorse for determining structural models in the field of biophysics. Within macromolecular crystallography, the molecular replacement method has been a highly successful method for providing phasing models to combine with experimental information to obtain 3D models. In

CHAPTER 7.

this dissertation we demonstrate that an alternative to molecular replacement, called “phasing by packing”, is promising for multi-rigid-domain structures. Numerical results with toy models illustrate the potential of this method in the context of both the planar wallpaper symmetry $p1$, and space-group symmetries, $P1$ and $P2_12_12_1$. Future work will apply this methodology to real problems.

- A new sampling method for the rotation group $SO(3)$ and the Euclidean group $SE(2)$ is proposed. In this method, those Lie groups are partitioned into Voronoi cells based on the elements of their discrete subgroups. Uniform Cartesian grids in exponential coordinates are generated on the Voronoi cell centered on the identity, and samples on the surrounding Voronoi cells are then replicated through shifting by the rotational symmetry operations. This sampling method can achieve almost-uniform sampling at any level of resolution, without having to store large numbers of coordinates or requiring sophisticated data structures. In the implementation for the protein crystal packing problem, the sampling method shows efficiency in finding collision-free conformations, and it also has great potentials in the robot motion planning problem.
- We also present an approach to parameterizing the exact boundaries of the Minkowski sum and difference of two ellipsoids. We develop two other methods to quickly obtain the approximated Minkowski sum and difference based on the offset curve/surface approximations and special properties of the Gaussian distribution. In contrast to most existing methods, our approaches are completely analytical and have closed

CHAPTER 7.

forms. Therefore they naturally provide improved efficiency and better robustness in many applications. With our exact parameterization, the volumes of the Minkowski sum and difference of two ellipsoids can be calculated efficiently and numerically, but in general, formulas for the volumes enclosed in these regions do not have exact closed-form expressions. Based on Steiner's Formula, we also develop a method to provide the upper and lower bounds of the Minkowski-sum volumes. These bounds are within a few percent of the actual values over a wide range of aspect ratios. In the last part of the paper, we also introduce a containment formula based on the Principal Kinematic Formula to compute the sum of all volumes of the Minkowski difference of two ellipsoids when one can move freely at all orientations inside another without collision. Based on this closed-form representation of the Minkowski sum and difference of two ellipsoids, 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. For articulated bodies, a hybrid approach integrating with existing sampling-based approaches with a "safety function" based on our prior knowledge of collision-free regions for single-rigid-bodies is proposed. For both scenarios, we have demonstrated the efficacy and the potential of our method with both planar and spatial examples, especially for the

CHAPTER 7.

narrow passage problem. Future work will be concerned with hardening code and applying these methodologies to various real-world problems in protein crystallography and robotics.

Appendix A

Matlab code for space group calculations

A.1 Matlab Code for Normal Subgroup Test

```
function Main_CompLeftRightCosets

% MAIN_COMPLEFTRIGHTCOSETS Normal subgroup test
%
% Description:
%   Main function: Check if each representative (rep) in the left coset
%   space of a subgroup is equivalent to a rep in the right coset of
%   the subgroup. If true, display 'The subgroup is normal', and if
%   false, display 'The subgroup is not normal'.
%
% Features:
%   - Import the txt files that contains the information of the left
%   and right coset spaces, e.g., 'Left_Coset.txt' and
%   'Right_Coset.txt'.
%   - A left and a right coset reps are equivalent to each other iff
%   the left coset rep is equivalent to the right coset under
%   the modulo P1 arithmetic.
```

APPENDIX A.

```

%      - Return the matching results of the left and right coset spaces.
%      - Display 'The subgroup is not normal' when two left coset
%      representatives are equivalent to the same right coset rep.
%
% Author:
%      Yan Yan
%
% Last Updated:
%      February 2013

% -----
clc; close all;
% Read in files and save the coset reps into strings
Name_L = 'Left_Coset';
Name_R = 'Right_Coset';
data_L = textread([sprintf('%s', Name_L), '.txt'], '%s', 'delimiter', '');
data_R = textread([sprintf('%s', Name_R), '.txt'], '%s', 'delimiter', '');
idx = strfind(data_L, 'Coset');
n = sum(cell2mat(idx)); % # of coset reps in the coset space
m = (length(data_L))/n-1; % # of elements in each coset rep

% Check if each rep in the left coset space has a corresponding rep in the
% right coset space.
%
% If one rep in the left coset space does not have a
% corresponding one in the right coset space, discontinue checking and
% report that the subgroup is not normal. Since the rest reps will not be
% checked, the corresponding entries in the matching results, i.e., I_R2L,
% will be all zeros.

val_j = 1:n;
I_LRSame = 1;
I_R2L = zeros(1, n);
for i = 1:n
    coset_L = data_L(2+(i-1)*(m+1): i*(m+1));
    for j = val_j
        coset_R = data_R(2+(j-1)*(m+1): j*(m+1));
        if CosetsComp(coset_L, coset_R)
            I_R2L(i) = j;
            I_j = find(val_j == j);
            val_j(I_j) = [];
            break
        end
    end
end

```

APPENDIX A.

```

        end
    end
    if ~I_R2L(i)
        I_LRSame = 0;
        disp('The subgroup is not normal.')
```

```

        break
    end
    progressbar(i/n)
end
close all;
I_R2L
if I_LRSame
    disp('The subgroup is normal.')
```

```

end

% =====

function I = CosetsComp(coset_L, coset_R)

% COSETSCOMP Compare a left coset rep with a right coset rep
%
% I = CosetsComp(coset_L, coset_R) check if a left coset rep is equivalent
% to a right coset rep (under the modulo P1)
%
% Argument Description:
%     coset_L - the left coset rep
%     coset_R - the right coset rep
%
% Output Description:
%     I - '0' if the left and right coset reps are
%     equivalent, and '1' if they are not equivalent
%
% Author:
%     Yan Yan
%
% Last Updated:
%     February 2013

% -----
I = 1; % initialize I
m = size(coset_L, 1);
I_oper_R2L = zeros(1, m); % # of elements in each coset rep
```

APPENDIX A.

```

for i = 1:m
    operation_L = char(coset_L(i));
    operation_L = operation_L(2: length(operation_L)-1);
%   find the Coefficients matrix A and b for the left coset rep
[A_oper_L, b_oper_L] = ReadCoefficients_WO_mod(operation_L);
for j = 1:m
    operation_R = char(coset_R(j));
    operation_R = operation_R(2: length(operation_R)-1);
%   find the Coefficients matrix A and b for the left coset rep
[A_oper_R, b_oper_R] = ReadCoefficients_WO_mod(operation_R);
if isequal(A_oper_L, A_oper_R) &&...
    isequal(mod(b_oper_R-b_oper_L, 1), zeros(3,1))
    I_oper_R2L(i) = j;
    break
end
end
if ~I_oper_R2L(i)
%   disp('No correspondence in this pair!')
    I = 0;
    break
end
end

% =====

function [A, b] = ReadCoefficients_WO_mod(Coset)

% READCOEFFICIENTS_WO_MOD Extract the coefficient matrices of a coset rep
%
% [A, b] = ReadCoefficients_WO_mod(Coset) returns the coefficient matrices
% A and b of a given coset rep
%
% Example:
%   Coset = char('-2x+1/4+y+z,-x+y+z+5/4,-x+y-z+1/4');
%   [A, b] = ReadCoefficients_WO_mod(Coset)
%   A =
%       -2     1     1
%       -1     1     1
%       -1     1    -1
%   b =
%       0.2500

```

APPENDIX A.

```

%           1.2500
%           0.2500
%
% Author:
%       Yan Yan
%
% Last Updated:
%       February 2013

% -----
n = length(Coset);
A = zeros(3, 3);
b = zeros(3, 1);
% Devide the char into a triplet
I_comma = strfind(Coset, ','); % Find the location indices of ','
I_start = [1 I_comma(1)+1 I_comma(2)+1]; % Start indices of the triplets
I_end = [I_comma(1)-1 I_comma(2)-1 n]; % End indices of the triplets

for i = 1:3
    coset = Coset(I_start(i): I_end(i));
    % make sure there is a "sign" at the beginning of each triplet
    if coset(1) ~= '-'
        coset = ['+' coset];
    end
    I_plus = strfind(coset, '+');
    I_minus = strfind(coset, '-');
    I_x = strfind(coset, 'x');
    I_y = strfind(coset, 'y');
    I_z = strfind(coset, 'z');
    I_node = sort([I_plus, I_minus, I_x, I_y, I_z]);
    const = coset;
    if I_x
        if coset(I_x-1) == '-'
            A(i, :) = A(i, :) + [-1 0 0];
        elseif coset(I_x-1) == '+'
            A(i, :) = A(i, :) + [1 0 0];
        else
            A(i, :) = A(i, :) + ...
                [str2num(coset(I_node(find(I_node==I_x)-1):I_x-1)), 0, 0];
        end
        const( strfind(const, coset(I_node(find(I_node==I_x)-1):I_x)) :...
            strfind(const, coset(I_node(find(I_node==I_x)-1):I_x))+ ...

```

APPENDIX A.

```

        I_x- I_node(find(I_node==I_x)-1) )=[];
    end

    if I_y
        if coset(I_y-1) == '-'
            A(i, :) = A(i, :) + [0 -1 0];
        elseif coset(I_y-1) == '+'
            A(i, :) = A(i, :) + [0 1 0];
        else
            A(i, :) = A(i, :) + ...
                [0 str2num(coset(I_node(find(I_node==I_y)-1):I_y-1)) 0];
        end
        const( strfind(const, coset(I_node(find(I_node==I_y)-1):I_y)) : ...
            strfind(const, coset(I_node(find(I_node==I_y)-1):I_y))+ ...
            I_y- I_node(find(I_node==I_y)-1) )=[];
    end

    if I_z
        if coset(I_z-1) == '-'
            A(i, :) = A(i, :) + [0 0 -1];
        elseif coset(I_z-1) == '+'
            A(i, :) = A(i, :) + [0 0 1];
        else
            A(i, :) = A(i, :) + ...
                [0 0 str2num(coset(I_node(find(I_node==I_z)-1):I_z-1))];
        end
        const( strfind(const, coset(I_node(find(I_node==I_z)-1):I_z)) : ...
            strfind(const, coset(I_node(find(I_node==I_z)-1):I_z))+ ...
            I_z- I_node(find(I_node==I_z)-1) )=[];
    end
    end
    if ~isempty(const)
        b(i) = str2num(const);
    end
end
end

```

A.2 Matlab Code for the Graphical User Interface (GUI) of Group Calculations

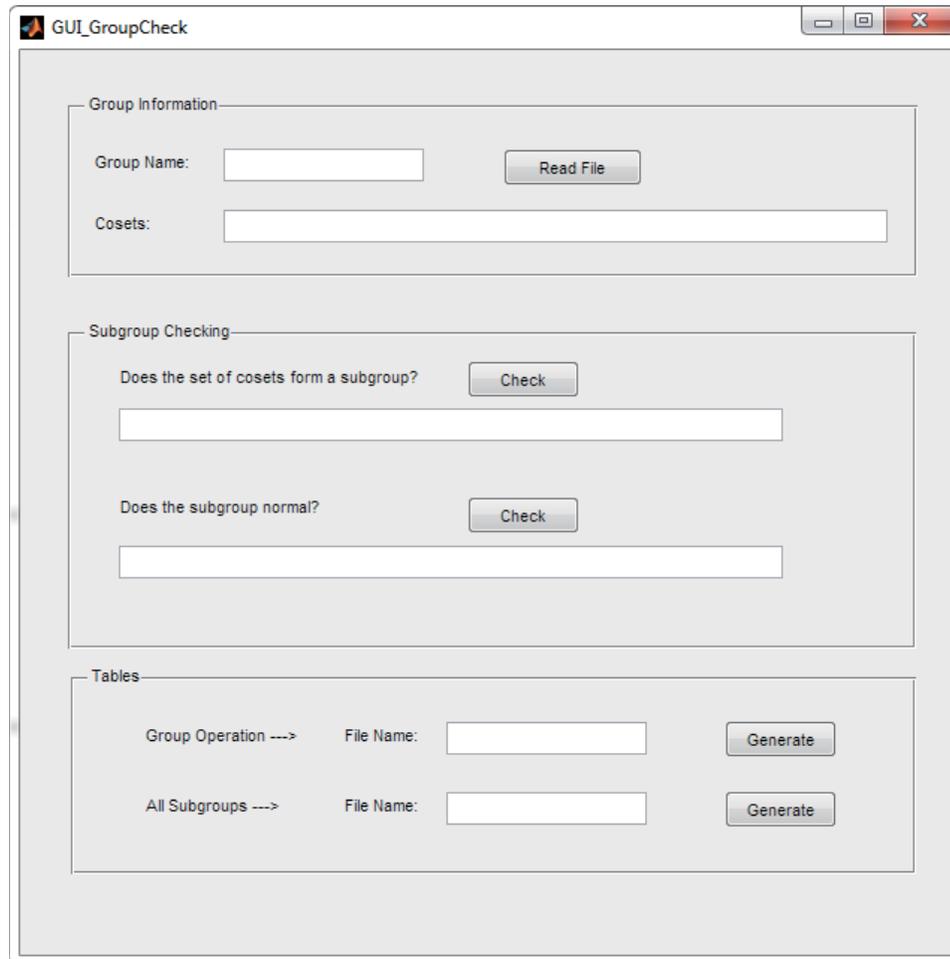


Figure A.1: The GUI of group calculations. The GUI can provide four basic functions — subgroup test, normal subgroup test, group table generation and subgroup table generation.

```
function varargout = GUI_GroupCheck(varargin)
% GUI_GROUPCHECK M-file for GUI_GroupCheck.fig
%   GUI_GROUPCHECK creates a GUI for some basic space group
%   calculations --- subgroup test, normal subgroup test, group table
%   generation and subgroup table generation.
%
```

APPENDIX A.

```
% Features:
%   - Import the txt files that contains the coset reps of  $G \setminus P_1$ , where
%   G is the space group to be checked.
%   - By providing the indices of some coset reps, the program can
%   check if these coset reps form a subgroup and if true, it can
%   further check if the resulting subgroup is normal.
%   - The program can generate group operation table, i.e., a n by n
%   matrix, where n is the number of the coset reps in  $G \setminus P_1$ . Each
%   (i, j) entry of this matrix corresponds to the index of resulting
%   coset rep from the group operation  $g_i \circ g_j$ . The results are saved
%   in an excel spreadsheet.
%   - The program can find all the possible subgroups in G, with the
%   indices of coset reps in each subgroup saved in a single row. The
%   results are also saved in an Excel spreadsheet.
%
% Author:
%   Yan Yan
%
% Last Updated:
%   February 2013

% -----
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @GUI_GroupCheck_OpeningFcn, ...
                  'gui_OutputFcn',  @GUI_GroupCheck_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI_GroupCheck is made visible.
function GUI_GroupCheck_OpeningFcn(hObject, eventdata, handles, varargin)
```

APPENDIX A.

```
handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_GroupCheck_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
clc
set(handles.edit_GroupName, 'String', '')
set(handles.edit_Cosets, 'String', '')
set(handles.edit_SubgroupCheck, 'String', '')
set(handles.edit_NormalSubgroupCheck, 'String', '')
set(handles.edit_FileName_GroupOperations, 'String', '')
set(handles.edit_FileName_AllSubgroups, 'String', '')

% --- Executes on button press in pushbutton_Subgroup.
function pushbutton_Subgroup_Callback(hObject, eventdata, handles)
Name = get(handles.edit_GroupName, 'String');
Cosets = get(handles.edit_Cosets, 'String');
Cosets = str2num(Cosets);
Display = Function_SubgroupCheck(Name, Cosets);
set(handles.edit_SubgroupCheck, 'String', Display)

% --- Executes on button press in pushbutton_NormalSubgroup.
function pushbutton_NormalSubgroup_Callback(hObject, eventdata, handles)
Name = get(handles.edit_GroupName, 'String');
Cosets = get(handles.edit_Cosets, 'String');
Cosets = str2num(Cosets);
Display = Function_NormalSubgroupCheck(Name, Cosets);
set(handles.edit_NormalSubgroupCheck, 'String', Display)

function edit_GroupName_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit_GroupName_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'), ...
    get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit_Cosets_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
```

APPENDIX A.

```
function edit_Cosets_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton_ReadGroupFile.
function pushbutton_ReadGroupFile_Callback(hObject, eventdata, handles)
Name = get(handles.edit_GroupName,'String');
Function_ReadGroupFile(Name);

function edit_SubgroupCheck_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit_SubgroupCheck_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_NormalSubgroupCheck_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit_NormalSubgroupCheck_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_FileName_GroupOperations_Callback(hObject,eventdata,handles)

% --- Executes during object creation, after setting all properties.
function edit_FileName_GroupOperations_CreateFcn(hObject,eventdata,handles)

if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton_GroupOperations.
function pushbutton_GroupOperations_Callback(hObject, eventdata, handles)
Name = get(handles.edit_GroupName, 'String');
```

APPENDIX A.

```
set(handles.edit_FileName_GroupOperations, 'String', ...
    ['Table_', sprintf('%s', Name), '.xls'])
Function_GroupOperations(Name)

function edit_FileName_AllSubgroups_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit_FileName_AllSubgroups_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton_AllSubgroups.
function pushbutton_AllSubgroups_Callback(hObject, eventdata, handles)
Name = get(handles.edit_GroupName, 'String');
Function_AllSubgroups(Name)
set(handles.edit_FileName_AllSubgroups, 'String',...
    ['Subgroups_', sprintf('%s', Name), '.xls'])

% =====

function Function_ReadGroupFile(Name)
% FUNCTION_READGROUPFILE Read the group file
%     Function_ReadGroupFile(Name) extracts the coefficients (A, b) of
%     each coset rep of the given group file and finds the inverse of
%     each coset rep by saving the index of the inverse coset rep, i.e.,
%     Ind_inv.
%
% Argument Description:
%     Name - the name of the txt file that contains all the coset reps
%     information.
%
% Output Description:
%     Save A, b and Ind_inv in the M-file with the same name as the
%     original txt file, under the current directory.
% Author:
%     Yan Yan
%
% Last Updated:
%     February 2013
```

APPENDIX A.

```

% -----
data = textread([sprintf('%s', Name), '.txt'], '%s', 'delimiter', '');
n = length(data)/2; % the odd line: 'Coset 1'; the even line: '(x, y, z)'
A = zeros(3, 3, n); b = zeros(3, n);

syms x y z
%===== find the coefficients of each coset (i.e., A, b) =====
for i = 1:n
    coset = char(data{2*i});
    coset = coset(2: length(coset)-1);
    [A_coset, b_coset] = ReadCoefficients(coset);
    A(:, :, i) = A_coset;
    b(:, i) = b_coset;
end

%===== find the inverse of each coset =====
Ind_inv = zeros(1, n);
for i = 1:n
    fx = A(1, :, i)*[x; y; z] + b(1, i); % the second multiplier
    fy = A(2, :, i)*[x; y; z] + b(2, i);
    fz = A(3, :, i)*[x; y; z] + b(3, i);
    for j = 1:n
        hx = A(1, :, j)*[fx; fy; fz] + b(1, j); % the first multiplier
        hy = A(2, :, j)*[fx; fy; fz] + b(2, j);
        hz = A(3, :, j)*[fx; fy; fz] + b(3, j);
        if all( ([hx, hy, hz]-[x, y, z]) == [0, 0, 0]) || ...
            all( abs([hx, hy, hz]-[x, y, z]) == [0, 0, 1]) ...
            || all( abs([hx, hy, hz]-[x, y, z]) == [0, 1, 1]) ...
            || all( abs([hx, hy, hz]-[x, y, z]) == [0, 1, 0])...
            || all( abs([hx, hy, hz]-[x, y, z]) == [1, 0, 0]) ...
            || all( abs([hx, hy, hz]-[x, y, z]) == [1, 0, 1]) ...
            || all( abs([hx, hy, hz]-[x, y, z]) == [1, 1, 1]) ...
            || all( abs([hx, hy, hz]-[x, y, z]) == [1, 1, 0]);
            Ind_inv(i) = j;
            break
        end
    end
    progressbar(i/n)
end
save(sprintf('%s', Name), 'A', 'b', 'Ind_inv')

% =====

```

APPENDIX A.

```

function Display = Function_SubgroupCheck(Name, I_cand)
% FUNCTION_SUBGROUPCHECK Subgroup test
%     Display = Function_SubgroupCheck(Name, I_cand) checks if the given
%     coset reps form a subgroup and display the result.
%
% Argument Description:
%     Name - the name of the txt file that contains all the coset reps
%     information.
%     I_cand - the indices of the coset reps that are going to check.
%
% Output Description:
%     Display the checking result. If true, display 'Yes, it is a
%     subgroup'. If false, display 'No, it is not a subgroup' and also
%     provide an counter-example.
% Author:
%     Yan Yan
%
% Last Updated:
%     February 2013

% -----
syms x y z fx fy fz hx hy hz
d = load(sprintf('%s', Name));
A = d. A;
b = d. b;
n = size(A, 3);
m = length(I_cand);
C_subgroup = zeros(m, m);

for i = 1:m
    for j = 1:m
        fx = A(1, :, I_cand(i))*[x; y; z] + b(1, I_cand(i));
        fy = A(2, :, I_cand(i))*[x; y; z] + b(2, I_cand(i));
        fz = A(3, :, I_cand(i))*[x; y; z] + b(3, I_cand(i));
        hx = A(1, :, I_cand(j))*[fx; fy; fz] + b(1, I_cand(j));
        hy = A(2, :, I_cand(j))*[fx; fy; fz] + b(2, I_cand(j));
        hz = A(3, :, I_cand(j))*[fx; fy; fz] + b(3, I_cand(j));
        hx = collect(hx, [x y z]);
        hy = collect(hy, [x y z]);
        hz = collect(hz, [x y z]);
        hx = char(hx); hx = hx(hx~= ' ');
    end
end

```

APPENDIX A.

```

    hy = char(hy); hy = hy(hy~= ' ');
    hz = char(hz); hz = hz(hz~= ' ');
    Ans = [hx, char(',') , hy, char(',') , hz];
    [A_Ans, b_Ans] = ReadCoefficients(Ans);
    for k = 1:n
        if isequal(A_Ans, A(:, :, k)) && isequal(b_Ans, b(:, k))
            C_subgroup(i, j) = k;
            continue
        end
    end
    end
    end
    progressbar(i/m)
end
I = ismember(C_subgroup,I_cand);
if isequal(I, ones(m ,m))
    Display = 'Yes, it is a subgroup.';
else
    [I_row, I_col] = find(I == 0);
    c_false = C_subgroup(I_row(1), I_col(1));
    Display = ['No, it is not a subgroup, since r_', ...
        sprintf('%s', num2str(I_col(1))), '*r_', ...
        sprintf('%s', num2str(I_cand(I_row(1))))), '*r_', ...
        sprintf('%s', num2str(I_col(1))), '^{-1}=', '*r_', ...
        sprintf('%s', num2str(c_false)), '.' ] ;
end

% =====

function Display = Function_NormalSubgroupCheck(Name, I_cand)
% FUNCTION_NORMALSUBGROUPCHECK Normal subgroup test
%     Display = Function_NormalSubgroupCheck(Name, I_cand) checks if
%     the given coset reps form a normal subgroup and display the result.
%
% Argument Description:
%     Name - the name of the txt file that contains all the coset reps
%     information.
%     I_cand - the indices of the coset reps that are going to check.
%
% Output Description:
%     Display the checking result. If true, display 'Yes, it is a normal
%     subgroup'. If false, display 'No, it is not a normal subgroup' and
%     also provide an counter-example.

```

APPENDIX A.

```

% Author:
%     Yan Yan
%
% Last Updated:
%     February 2013

% -----
syms x y z fx fy fz hx hy hz gx gy gz

d = load(sprintf('%s', Name));

A = d. A;
b = d. b;
Ind_inv = d. Ind_inv;
n = size(A, 3);
m = length(I_cand);
C_normal = zeros(m, n);
for k = 1:m
    for i = 1:n
        fx = A(1, :, Ind_inv(i))*[x; y; z] + b(1, Ind_inv(i));
        fy = A(2, :, Ind_inv(i))*[x; y; z] + b(2, Ind_inv(i));
        fz = A(3, :, Ind_inv(i))*[x; y; z] + b(3, Ind_inv(i));
        hx = A(1, :, I_cand(k))*[fx; fy; fz] + b(1, I_cand(k));
        hy = A(2, :, I_cand(k))*[fx; fy; fz] + b(2, I_cand(k));
        hz = A(3, :, I_cand(k))*[fx; fy; fz] + b(3, I_cand(k));
        gx = A(1, :, i)*[hx; hy; hz] + b(1, i);
        gy = A(2, :, i)*[hx; hy; hz] + b(2, i);
        gz = A(3, :, i)*[hx; hy; hz] + b(3, i);
        gx = collect(gx, [x y z]);
        gy = collect(gy, [x y z]);
        gz = collect(gz, [x y z]);
        gx = char(gx); gx = gx(gx~= ' ');
        gy = char(gy); gy = gy(gy~= ' ');
        gz = char(gz); gz = gz(gz~= ' ');
        Ans = [gx, char(',') , gy, char(',') , gz];
        [A_Ans, b_Ans] = ReadCoefficients(Ans);
        for j = 1:n
            if isequal(A_Ans, A(:, :, j)) && isequal(b_Ans, b(:, j))
                C_normal(k, i) = j;
                continue
            end
        end
    end
end

```

APPENDIX A.

```

    end
    progressbar(k/m)
end
I = ismember(C_normal,I_cand);
if isequal(I, ones(m ,n))
    Display = 'Yes, it is a normal subgroup.';
else
    [I_row, I_col] = find(I == 0);
    c_false = C_normal(I_row(1), I_col(1));
    Display = ['No, it is not a normal subgroup, since r_', ...
        sprintf('%s', num2str(I_col(1))), '*r_', ...
        sprintf('%s', num2str(I_cand(I_row(1))))), '*r_', ...
        sprintf('%s', num2str(I_col(1))), '^{-1}=', '*r_', ...
        sprintf('%s', num2str(c_false)), '.' ]];
end

% =====

function Function_GroupOperations(Name)
% FUNCTION_GROUPOPERATIONS Generate Group Table
%     Function_GroupOperations(Name) generates group operation table,
%     i.e., a n by n matrix, where n is the number of the coset reps in
%     G\P1. Each (i, j) entry of this matrix corresponds to the index of
%     resulting coset rep from the group operation g_i o g_j.
%
% Argument Description:
%     Name - the name of the txt file that contains all the coset reps
%     information.
%
% Output Description:
%     Save indices of coset reps resulting from the group operations
%     between all pairs of coset reps in an Excel file with the file name
%     'Table_Name', under the current directory.
% Author:
%     Yan Yan
%
% Last Updated:
%     February 2013

% -----
syms x y z fx fy fz hx hy hz
d = load(sprintf('%s', Name));

```

APPENDIX A.

```

A = d. A;
b = d. b;
n = size(A, 3);
I_cand = 1:n;
m = length(I_cand);
C_subgroup = zeros(m, m);

for i = 1:m
    for j = 1:m
        fx = A(1, :, I_cand(i))*[x; y; z] + b(1, I_cand(i));
        fy = A(2, :, I_cand(i))*[x; y; z] + b(2, I_cand(i));
        fz = A(3, :, I_cand(i))*[x; y; z] + b(3, I_cand(i));
        hx = A(1, :, I_cand(j))*[fx; fy; fz] + b(1, I_cand(j));
        hy = A(2, :, I_cand(j))*[fx; fy; fz] + b(2, I_cand(j));
        hz = A(3, :, I_cand(j))*[fx; fy; fz] + b(3, I_cand(j));
        hx = collect(collect(hx, [x y z]), [y z x]);
        hy = collect(collect(hy, [x y z]), [y z x]);
        hz = collect(collect(hz, [x y z]), [y z x]);
        hx = char(hx); hx = hx(hx~= ' ');
        hy = char(hy); hy = hy(hy~= ' ');
        hz = char(hz); hz = hz(hz~= ' ');
        Ans = [hx, char(',') , hy, char(',') , hz];
        [A_Ans, b_Ans] = ReadCoefficients(Ans);
        for k = 1:n
            if isequal(A_Ans, A(:, :, k)) && isequal(b_Ans, b(:, k))
                C_subgroup(i, j) = k;
                continue
            end
        end
        end
        end
        end
        progressbar(i/m)
    end
end
save(['Table_', sprintf('%s', Name)], 'C_subgroup');
xlswrite(['Table_', sprintf('%s', Name)], C_subgroup)

% =====

function Function_AllSubgroups(Name)
% FUNCTION_ALLSUBGROUPS Find all possible subgroups in a given group
% Function_AllSubgroups(Name) finds all the possible subgroups in a
% given group, with the indices of coset reps in each subgroup saved
% in a single row. The results are also saved in an Excel spreadsheet.

```

APPENDIX A.

```
%
% Argument Description:
%     Name - the name of the txt file that contains all the coset reps
%     information.
%
% Output Description:
%     Save the indices of all the subgroups in an Excel file with the
%     file name 'Subgroups_Name', under the current directory.
% Author:
%     Yan Yan
%
% Last Updated:
%     February 2013

% -----
d = load(sprintf('%s', Name));
A = d. A;
b = d. b;
n = size(b, 2);

T = load(['Table_', sprintf('%s', Name)]);
C_subgroup = T.C_subgroup;

t = 0;
I_cand = 1+t : n+t;
C = C_subgroup(I_cand, I_cand);
% all the combinations
    I_subgroup = [];
    m = n-1;
for k = 1: m
    combos = combntns(1:n, k);
    for i = 1:length(combos)
        I = ismember(C(combos(i, :), combos(i, :)), combos(i, :)+t);
        if isequal(I, ones(k,k))
            I_subgroup = [I_subgroup; combos(i, :)+t zeros(1, n-k)];
        end
    end
    end
    progressbar(k/m)
end
k = n;
I = ismember(C(1:k, 1:k), 1+t:k+t);
if isequal(I, ones(k,k))
```

APPENDIX A.

```
        I_subgroup = [I_subgroup; 1+t:k+t zeros(1, n-k)];
    end
    progressbar(k/n)

    save(['Subgroups_', sprintf('%s', Name)], 'I_subgroup')
    xlswrite(['Subgroups_', sprintf('%s', Name)], I_subgroup)
```

Bibliography

- [1] M. Khatib, H. Jaouni, R. Chatila, and J.-P. Laumond, “Dynamic path modification for car-like nonholonomic mobile robots,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, pp. 2920–2925, IEEE, 1997.
- [2] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.
- [3] O. B. Bayazit, G. Song, and N. M. Amato, “Enhancing randomized motion planners: Exploring with haptic hints,” *Autonomous Robots*, vol. 10, no. 2, pp. 163–174, 2001.
- [4] H. Chang and T.-Y. Li, “Assembly maintainability study with motion planning,” in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 1, pp. 1012–1019, IEEE, 1995.
- [5] Y. Koga, K. Kondo, J. Kuffner, and J.-C. Latombe, “Planning motions with intentions,” in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 395–408, ACM, 1994.

BIBLIOGRAPHY

- [6] J. Kuffner Jr and J.-C. Latombe, “Interactive manipulation planning for animated characters,” in *Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on*, pp. 417–418, IEEE, 2000.
- [7] A. P. Singh, J.-C. Latombe, and D. L. Brutlag, “A motion planning approach to flexible ligand binding,” in *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pp. 252–261, AAAI Press, 1999.
- [8] N. M. Amato and G. Song, “Using motion planning to study protein folding pathways,” *Journal of Computational Biology*, vol. 9, no. 2, pp. 149–168, 2002.
- [9] Y. Yan and G. S. Chirikjian, “Molecular replacement for multi-domain structures using packing models,” in *Proceedings of the ASME 2011 IDETC*, ASME, 2011.
- [10] Y. Yan and G. S. Chirikjian, “A Gaussian packing model for phasing in macromolecular crystallography,” in *BIOCOMP*, 2011.
- [11] Y. Yan and G. S. Chirikjian, “Packing models for multi-domain biomolecular structures in crystals with p212121 space-group symmetry,” in *Workshops at the 20th AAAI Conference on Artificial Intelligence*, 2013.
- [12] G. S. Chirikjian and Y. Yan, “Mathematical aspects of molecular replacement. II. geometry of motion spaces,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 68, no. 2, pp. 208–221, 2012.
- [13] Y. Shiao, J.-N. Wang, Y.-F. Huang, and C.-K. Huang, “EMPSC: A new method based

BIBLIOGRAPHY

- on ellipsoidal model for protein structure comparison,” *Technique report, Department of Engineering Science and Ocean Engineering, National Taiwan University*, 2006.
- [14] V. J. Lumelsky and A. A. Stepanov, “Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape,” *Algorithmica*, vol. 2, no. 1-4, pp. 403–430, 1987.
- [15] I. Kamon, E. Rivlin, and E. Rimon, “A new range-sensor based globally convergent navigation algorithm for mobile robots,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 429–435, IEEE, 1996.
- [16] T. Lozano-Pérez and M. A. Wesley, “An algorithm for planning collision-free paths among polyhedral obstacles,” *Communications of the ACM*, vol. 22, pp. 560–570, Oct. 1979.
- [17] T. Lozano-Perez, “Spatial planning: A configuration space approach,” *IEEE Transactions on Computers*, vol. C-32, pp. 108–120, Feb. 1983.
- [18] R. Brooks, “Solving the find-path problem by good representation of free space,” *IEEE Trans. Syst., Man, Cybern.*, vol. 13, no. 3, pp. 190 – 197, 1983.
- [19] T. Lozano-Perez, “A simple motion-planning algorithm for general robot manipulators,” *IEEE Journal on Robotics and Automation*, vol. 3, pp. 224–238, June 1987.
- [20] L. Kavraki, “Computation of configuration-space obstacles using the fast Fourier

BIBLIOGRAPHY

- transform,” *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 408–413, June 1995.
- [21] A. B. Kyatkin and G. S. Chirikjian, “Algorithms for fast convolutions on motion groups,” *Applied and Computational Harmonic Analysis*, vol. 9, no. 2, pp. 220–241, 2000.
- [22] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [23] J. Lengyel, M. Reichert, B. R. Donald, and D. P. Greenberg, “Real-time robot motion planning using rasterizing computer graphics hardware,” *ACM SIGGRAPH Computer Graphics*, vol. 24, pp. 327–335, Sept. 1990.
- [24] J. Barraquand, B. Langlois, and J.-C. Latombe, “Numerical potential field techniques for robot path planning,” in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, pp. 1012–1017 vol.2, IEEE, 1991.
- [25] D. E. Koditschek and E. Rimon, “Robot navigation functions on manifolds with boundary,” *Advances in Applied Mathematics*, vol. 11, no. 4, pp. 412–442, 1990.
- [26] E. Rimon and D. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

BIBLIOGRAPHY

- [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] F. Lamiroux and J. Laumond, “On the expected complexity of random path planning,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3014–3019, IEEE, 1996.
- [29] O. Brock and O. Khatib, “High-speed navigation using the global dynamic window approach,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, pp. 341–346, IEEE, 1999.
- [30] O. Takahashi and R. Schilling, “Motion planning in a plane using generalized Voronoi diagrams,” *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 143–150, Apr. 1989.
- [31] H. Choset and K. Nagatani, “Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization,” *IEEE Transactions on Robotics and Automation*, no. 2, pp. 125–137, 2001.
- [32] K. Hoff, T. Culver, J. Keyser, M. Lin, and D. Manocha, “Interactive motion planning using hardware-accelerated computation of generalized Voronoi diagrams,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 3, pp. 2931–2937, IEEE, 2000.

BIBLIOGRAPHY

- [33] S. Garrido, L. Moreno, M. Abderrahim, and F. Martin, "Path planning for mobile robot navigation using Voronoi diagram and fast marching," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2376–2381, IEEE, Oct. 2006.
- [34] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robotics and Autonomous Systems*, vol. 8, pp. 47–63, Nov. 1991.
- [35] H. Shatkay and L. P. Kaelbling, "Learning topological maps with weak local odometric information," *International Joint Conference on Artificial Intelligence*, vol. 15, pp. Shatkay, H., & Kaelbling, L. P. (1997, August). Le, 1997.
- [36] S. Thrun and J. steffen Gutmann, "Integrating topological and metric maps for mobile robot navigation: A statistical approach," *In Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence*, no. 1, pp. 989–996, 1998.
- [37] S. Koenig and R. G. Simmons, "Xavier: A robot navigation architecture based on partially observable markov decision process models," *Artificial Intelligence Based Mobile Robotics Case Studies of Successful Robot Systems*, pp. 1–24, 1998.
- [38] I. Nourbakhsh, R. Powers, and S. Birchfield "DERVISH: An office-navigating robot," *AI magazine*, vol. 16, p. 53, June 1995.

BIBLIOGRAPHY

- [39] J. Canny, “Constructing roadmaps of semi-algebraic sets I: Completeness,” *Artificial Intelligence*, vol. 37, pp. 203–222, Dec. 1988.
- [40] K. O. Arras and N. Tomatis, “Multisensor on-the-fly localization: Precision and reliability for applications,” *European Workshop on Advanced Mobile Robots*, vol. 34, no. 2-3, pp. 131–143, 2001.
- [41] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, “An efficient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 1, pp. 206–211, IEEE, 2003.
- [42] I. Stamos and P. Allen, “Integration of range and image sensing for photo-realistic 3D modeling,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, pp. 1435–1440, IEEE, 2000.
- [43] S. Thrun, W. Burgard, and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, pp. 321–328, IEEE, 2000.
- [44] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics*

BIBLIOGRAPHY

- and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, pp. 146–151, IEEE Comput. Soc. Press, 1997.
- [45] A. Elfes, “Sonar-based real-world mapping and navigation,” *IEEE Journal on Robotics and Automation*, vol. 3, pp. 249–265, June 1987.
- [46] L. Kavraki, P. Svestka, J.-C. Latombe, and M. 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.
- [47] N. M. Amato and O. B. Bayazit, “OBPRM: An obstacle-based prm for 3D workspaces,” *Preprocessing, I., and Roadmap Construction*, 1998.
- [48] R. Geraerts and M. Overmars, *Algorithmic Foundations of Robotics V*, vol. 7 of *Springer Tracts in Advanced Robotics*. Berlin, Heidelberg: Springer, 2004.
- [49] D. Hsu and R. Kindel, “Randomized kinodynamic motion planning with moving obstacles,” *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2000.
- [50] R. Bohlin and L. Kavraki, “Path planning using lazy PRM,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, pp. 521–528, IEEE, 2000.
- [51] C. Nissoux, T. Simeon, and J.-P. Laumond, “Visibility based probabilistic

BIBLIOGRAPHY

- roadmaps,” in *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*, vol. 3, pp. 1316–1321, IEEE, 1999.
- [52] G. Varadhan, “Star-shaped roadmaps - a deterministic sampling approach for complete motion planning,” *Robotics*, 2005.
- [53] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” *In*, vol. 129, no. 98-11, pp. 98–11, 1998.
- [54] S. LaValle and J. Kuffner, “Randomized kinodynamic planning,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, pp. 473–479, IEEE, 1999.
- [55] S. M. Lavalle and J. J. Kuffner, “Rapidly-exploring random trees: Progress and prospects,” *Algorithmic and Computational Robotics New Directions*, pp. 293–308, 2000.
- [56] J. Kuffner and S. LaValle, “RRT-connect: An efficient approach to single-query path planning,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, pp. 995–1001, IEEE, 2000.
- [57] J. Bruce and M. Veloso, “Real-time randomized path planning for robot navigation,”

BIBLIOGRAPHY

- in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, pp. 2383–2388, IEEE, 2002.
- [58] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 1: Analytic Methods and Modern Applications*, vol. 1. Boston: Birkhäuser, 2009.
- [59] G. S. Chirikjian and A. B. Kyatkin, *Engineering Applications of Noncommutative Harmonic Analysis*. Boca Raton, FL: CRC Press, 2000.
- [60] F. C. Park, *The optimal kinematic design of mechanisms*. PhD thesis, Harvard University, Cambridge, MA, 1991.
- [61] C. J. Bradley and A. P. Cracknell, *The mathematical theory of symmetry in solids: representation theory for point groups and space groups*. Oxford: Clarendon Press Oxford, 1972.
- [62] M. Aroyo and et al, “Representations of crystallographic space groups, commission on mathematical and theoretical crystallography,” (*online report*), 2010.
- [63] W. Miller, *Symmetry groups and their applications*, vol. 50. New York: Academic Press, 1973.
- [64] A. McPherson, *Introduction to macromolecular crystallography*. Hoboken, NJ: Wiley-Blackwell, 2011.
- [65] G. Rhodes, *Crystallography made crystal clear: a guide for users of macromolecular models*. New York: Academic press, 2010.

BIBLIOGRAPHY

- [66] B. Rupp, *Biomolecular crystallography*. London: Garland Science, 2009.
- [67] S. W. Wukovitz and T. O. Yeates, “Why protein crystals favour some space-groups over others,” *Nature Structural & Molecular Biology*, vol. 2, no. 12, pp. 1062–1067, 1995.
- [68] F. Aurenhammer, “Voronoi diagrams a survey of a fundamental geometric data structure,” *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.
- [69] M. Erwig and F. Hagen, “The graph Voronoi diagram with applications,” *Networks*, vol. 36, no. 3, pp. 156–163, 2000.
- [70] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*. New York: Wiley, 1992.
- [71] D. Reem, “An algorithm for computing Voronoi diagrams of general generators in general normed spaces,” in *Voronoi Diagrams, 2009. ISVD’09. Sixth International Symposium on*, pp. 144–152, IEEE, 2009.
- [72] W. A. Hendrickson and K. B. Ward, “A packing function for delimiting the allowable locations of crystallized macromolecules,” *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 5, pp. 778–780, 1976.
- [73] D. E. Williams, “Crystal packing of molecules,” *Science*, vol. 147, no. 3658, pp. 605–606, 1965.

BIBLIOGRAPHY

- [74] A. Damiani, E. Giglio, A. M. Liquori, and L. Mazzarell, “Calculation of crystal packing: a novel approach to the phase problem,” *Nature*, vol. 215, pp. 1161–1162, 1967.
- [75] M. G. Rossmann and D. M. Blow, “The detection of sub-units within the crystallographic asymmetric unit,” *Acta Crystallographica*, vol. 15, no. 1, pp. 24–31, 1962.
- [76] M. G. Rossmann, “Molecular replacement-historical background,” *Acta Crystallographica Section D: Biological Crystallography*, vol. 57, no. 10, pp. 1360–1366, 2001.
- [77] J. Navaza, “AMoRe: an automated package for molecular replacement,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 50, no. 2, pp. 157–163, 1994.
- [78] S. Bailey, “The CCP4 suite: programs for protein crystallography,” *Acta Crystallographica Section D*, vol. 50, pp. 760–763, 1994.
- [79] D. C. Jamrog, Y. Zhang, and G. N. Phillips Jr, “SOMoRe: a multi-dimensional search and optimization approach to molecular replacement,” *Acta Crystallographica Section D: Biological Crystallography*, vol. 59, no. 2, pp. 304–314, 2003.
- [80] J. I. Jeong, E. E. Lattman, and G. S. Chirikjian, “A method for finding candidate conformations for molecular replacement using relative rotation between domains of

BIBLIOGRAPHY

- a known structure,” *Acta Crystallographica Section D: Biological Crystallography*, vol. 62, no. 4, pp. 398–409, 2006.
- [81] R. A. Crowther and D. M. Blow, “A method of positioning a known molecule in an unknown crystal structure,” *Acta Crystallographica*, vol. 23, no. 4, pp. 544–548, 1967.
- [82] R. A. Crowther, “The fast rotation function,” *Int. Sci. Rev. Ser.*, vol. 13, pp. 173–178, 1972.
- [83] E. E. Lattman and W. E. Love, “A rotational search procedure for detecting a known molecule in a crystal,” *Acta Crystallographica Section B: Structural Crystallography and Crystal Chemistry*, vol. 26, no. 11, pp. 1854–1857, 1970.
- [84] A. Vagin and A. Teplyakov, “MOLREP: an automated program for molecular replacement,” *Journal of Applied Crystallography*, vol. 30, no. 6, pp. 1022–1025, 1997.
- [85] E. Lattman, “Use of the rotation and translation functions,” *Methods in Enzymology*, vol. 115, pp. 55–77, 1985.
- [86] S. M. LaValle, *Planning algorithms*. Cambridge: Cambridge university press, 2006.
- [87] G. Chirikjian and S. Zhou, “Metrics on motion and deformation of solid models,” *Journal of Mechanical Design*, vol. 120, p. 252, 1998.

BIBLIOGRAPHY

- [88] G. Baldwin, R. Mahony, and J. Trumpf, “A nonlinear observer for 6 DOF pose estimation from inertial and bearing measurements,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2237–2242, 2009.
- [89] A. M. Bloch, *Nonholonomic mechanics and control*, vol. 24. New York: Springer, 2003.
- [90] R. W. Brockett, “System theory on group manifolds and coset spaces,” *SIAM Journal on Control*, vol. 10, no. 2, pp. 265–284, 1972.
- [91] T. Duncan, “An estimation problem in compact Lie groups,” *Systems & Control Letters*, vol. 10, no. 4, pp. 257–263, 1988.
- [92] V. Jurdjevic and H. J. Sussmann, “Control systems on Lie groups,” *Journal of Differential equations*, vol. 12, no. 2, pp. 313–329, 1972.
- [93] J. Kwon, M. Choi, F. C. Park, and C. Chun, “Particle filterin on the Euclidean group: framework and applications,” *Robotica*, vol. 25, no. 06, pp. 725–737, 2007.
- [94] T. Lee, N. H. McClamroch, and M. Leok, “Optimal control of a rigid body using geometrically exact computations on $SE(3)$,” in *Proceedings of the IEEE Conference on Decision and Control*, pp. 2710–2715, IEEE, 2006.
- [95] R. Mahony, T. Hamel, and J.-M. Pflimlin “Nonlinear complementary filter on the special orthogonal group,” *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

BIBLIOGRAPHY

- [96] A. S. Willsky, *Some estimation problems on Lie groups*. New York: Springer, 1973.
- [97] A. S. Willsky, *Dynamical systems defined on groups: structural properties and estimation*. PhD thesis, Massachusetts Institute of Technology, 1973.
- [98] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems*. New York: Springer, 2004.
- [99] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. Boca Raton, FL: CRC press, 1993.
- [100] A. Makadia and K. Daniilidis, “Rotation recovery from spherical images without correspondences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1170–1175, 2006.
- [101] R. Tron, R. Vidal, and A. Terzis, “Distributed pose averaging in camera networks via consensus on $SE(3)$,” in *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pp. 1–10, 2008.
- [102] J. Avro, *Graphics Gems III, Chapter: Fast Random Rotation Matrices*. New York: Academic Press, 1992.
- [103] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*, vol. 1. Birkhäuser, 2011.
- [104] K. Shoemake, *Graphics Gems III, Chapter: Uniform Random Rotations*. New York: Academic Press, 1992.

BIBLIOGRAPHY

- [105] E. Bannai and R. Damerell, “Tight spherical designs I,” *Journal of the Mathematical Society of Japan (JMSJ)*, pp. 199–207, 1979.
- [106] N. J. Sloane, J. Conway, *et al.*, *Sphere packings, lattices and groups*, vol. 290. New York: Springer, 1999.
- [107] P. Delsarte, J.-M. Goethals, and J. J. Seidel, “Spherical codes and designs,” *Geometriae Dedicata*, vol. 6, no. 3, pp. 363–388, 1977.
- [108] W. Neutsch, *Coordinates*, vol. 1. Berlin: Walter de Gruyter, 1996.
- [109] B. Clare and D. Kepert, “The optimal packing of circles on a sphere,” *Journal of mathematical chemistry*, vol. 6, no. 1, pp. 325–349, 1991.
- [110] L. Fejes Tóth, “Stable packing of circles on the sphere,” *Structural Topology 1985 núm 11*, 1985.
- [111] D. Kottwitz, “The densest packing of equal circles on a sphere,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 47, no. 3, pp. 158–165, 1991.
- [112] T. Tarnai, “Spherical circle-packing in nature, practice and theory,” *Structural Topology*, no. 9, pp. 39–58, 1984.
- [113] S. L. Sobolev, “Cubature formulas on the sphere invariant under finite groups of rotations,” in *Selected Works of SL Sobolev*, pp. 461–466, Springer, 2006.

BIBLIOGRAPHY

- [114] S. Sobolev, “Theory of cubature formulas,” in *Selected Works of SL Sobolev*, pp. 491–511, New York: Springer, 2006.
- [115] G. Yang and I.-M. Chen, “Equivolumetric partition of solid spheres with applications to orientation workspace analysis of robot manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 22, no. 5, pp. 869–879, 2006.
- [116] E. E. Lattman, “Optimal sampling of the rotation function,” *Acta Crystallographica Section B: Structural Crystallography and Crystal Chemistry*, vol. 28, no. 4, pp. 1065–1068, 1972.
- [117] D. Hardin and E. Saff, “Discretizing manifolds via minimum energy points,” *Notices of the AMS*, vol. 51, no. 10, pp. 1186–1194, 2004.
- [118] E. B. Saff and A. B. Kuijlaars, “Distributing many points on a sphere,” *The Mathematical Intelligencer*, vol. 19, no. 1, pp. 5–11, 1997.
- [119] K. M. Gorski, E. Hivon, A. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann, “HEALPix: a framework for high-resolution discretization and fast analysis of data distributed on the sphere,” *The Astrophysical Journal*, vol. 622, no. 2, p. 759, 2005.
- [120] S. R. Lindemann, A. Yershova, and S. M. LaValle, “Incremental grid sampling strategies in robotics,” in *Algorithmic Foundations of Robotics VI*, pp. 313–328, New York: Springer, 2005.

BIBLIOGRAPHY

- [121] J. J. Kuffner, “Effective sampling and distance metrics for 3D rigid body path planning,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3993–3998, IEEE, 2004.
- [122] J. C. Mitchell, “Sampling rotation groups by successive orthogonal images,” *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 525–547, 2008.
- [123] X. Sun and Z. Chen, “Spherical basis functions and uniform distribution of points on spheres,” *Journal of approximation theory*, vol. 151, no. 2, pp. 186–207, 2008.
- [124] G. Wagner, “On a new method for constructing good point sets on spheres,” *Discrete & Computational Geometry*, vol. 9, no. 1, pp. 111–129, 1993.
- [125] A. Yershova and S. M. LaValle, “Deterministic sampling methods for spheres and $SO(3)$,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3974–3980, IEEE, 2004.
- [126] A. Yershova, S. Jain, S. M. LaValle, and J. C. Mitchell, “Generating uniform incremental grids on $SO(3)$ using the hopf fibration” *The International Journal of Robotics Research*, vol. 29, no. 7, pp. 801–812, 2010.
- [127] G. Chirikjian and S. Zhou, “Metrics on motion and deformation of solid models,” *Journal of Mechanical Design*, vol. 120, p. 252, 1998.
- [128] F. L. Williams, *Lectures on the Spectrum of L (Γ)*. Longman Scientific & Technical, 1991.

BIBLIOGRAPHY

- [129] G. S. Chirikjian and Y. Yan, “Mathematical aspects of molecular replacement. II. geometry of motion spaces,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 68, no. 2, pp. 208–221, 2012.
- [130] M. G. Rossmann, “Molecular replacement-historical background,” *Acta Crystallographica Section D: Biological Crystallography*, vol. 57, no. 10, pp. 1360–1366, 2001.
- [131] A. Vagin and A. Teplyakov, “Molecular replacement with MOLREP,” *Acta Crystallographica Section D: Biological Crystallography*, vol. 66, no. 1, pp. 22–25, 2009.
- [132] Y. Yan and G. S. Chirikjian, “Almost-uniform sampling of rotations for conformational searches in robotics and structural biology,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4254–4259, IEEE, 2012.
- [133] Y. Yan and G. S. Chirikjian, “Voronoi cells in Lie groups and coset decompositions: implications for optimization, integration, and Fourier analysis,” in *Proceedings of the IEEE Conference on Decision and Control*, IEEE, 2013.
- [134] D. Stoyan, W. S. Kendall, J. Mecke, and D. Kendall, *Stochastic geometry and its applications*, vol. 8. Chichester: Wiley, 1987.
- [135] J.-C. Latombe, *Robot motion planning*. Boston: Kluwer academic publishers, 1990.
- [136] D. Halperin, J.-C. Latombe, and R. H. Wilson, “A general framework for assembly

BIBLIOGRAPHY

- planning: The motion space approach,” *Algorithmica*, vol. 26, no. 3-4, pp. 577–601, 2000.
- [137] 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.
- [138] E. Behar and J.-M. Lien, “Dynamic Minkowski sum of convex shapes,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3463–3468, IEEE, 2011.
- [139] 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.
- [140] 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.
- [141] 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.
- [142] 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.

BIBLIOGRAPHY

- [143] J.-R. Sack and J. Urrutia, *Handbook of computational geometry*. Amsterdam: North-Holland, 1999.
- [144] C. L. Bajaj and M.-S. Kim, “Generation of configuration space obstacles: The case of moving algebraic curves,” *Algorithmica*, vol. 4, no. 1-4, pp. 157–172, 1989.
- [145] A. Kaul and R. T. Farouki, “Computing Minkowski sums of plane curves,” *International Journal of Computational Geometry & Applications*, vol. 5, no. 04, pp. 413–432, 1995.
- [146] I.-K. Lee, M.-S. Kim, and G. Elber, “Polynomial/rational approximation of Minkowski sum boundary curves,” *Graphical Models and Image Processing*, vol. 60, no. 2, pp. 136–165, 1998.
- [147] P. Goodey and W. Weil, “Intersection bodies and ellipsoids,” *Mathematika*, vol. 42, pp. 295–304, Dec. 1995.
- [148] 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.
- [149] S. Alfano and M. L. Greer, “Determining if two solid ellipsoids intersect,” *Journal of guidance, control, and dynamics*, vol. 26, no. 1, pp. 106–110, 2003.
- [150] K. Chan, W. W. Hager, S.-J. Huang, P. M. Pardalos, O. A. Prokopyev, and P. Parda-

BIBLIOGRAPHY

- los, *Multiscale Optimization Methods and Applications*, vol. 82 of *Nonconvex Optimization and Its Applications*. Boston: Kluwer Academic Publishers, 2006.
- [151] L. Ros, A. Sabater, and F. Thomas, “An ellipsoidal calculus based on propagation and fusion,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 32, no. 4, pp. 430–442, 2002.
- [152] A. Kurzhanskiĭ and I. Vályi, *Ellipsoidal calculus for estimation and control*. Laxenburg, Austria and Boston: Iiasa Research Center, 1997.
- [153] A. A. Kurzhanskiy and P. Varaiya, “Ellipsoidal Toolbox (ET),” in *Decision and Control, 2006 45th IEEE Conference*, pp. 1498–1503, 2006.
- [154] D. H. Lehmer, “Approximations to the area of an n-dimensional ellipsoid,” *Canadian Journal of Mathematics*, vol. 2, pp. 267–282, 1950.
- [155] M. Klamkin, “Elementary approximations to the area of n-dimensional ellipsoids,” *The American Mathematical Monthly*, vol. 78, no. 3, pp. 280–283, 1971.
- [156] H. Hadwiger, *Altes und Neues über konvexe Körper*. Basel: Birkhäuser, 1955.
- [157] I. Rivin, “Surface area and other measures of ellipsoids,” *Advances in Applied Mathematics*, vol. 39, pp. 409–427, Oct. 2007.
- [158] E. Behar and J.-M. Lien, “Dynamic Minkowski sum of convex shapes,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 3463–3468, IEEE, 2011.

BIBLIOGRAPHY

- [159] R. Pfiel, “Surface area inequalities for ellipsoids using Minkowski sums,” *Geometriae Dedicata*, vol. 28, pp. 171–179, Nov. 1988.
- [160] D. Salomon, *Curves and surfaces for computer graphics*. New York: Springer, 2006.
- [161] “<http://www.numericana.com/answer/ellipsoid.htm#thomsen>,” 2006.
- [162] W. Blaschke, “Einige bemerkungen über kurven und fächen konstanter breite,” *Leipziger Berichte*, vol. 57, pp. 290–297, 1915.
- [163] W. Blaschke, *Vorlesungen über Integralgeometrie*. VEB Dt. Verlag d. Wiss., 1955.
- [164] H. Poincaré, *Calcul de Probabilités, 2nd ed.* Paris, 1912.
- [165] L. A. Santaló, *Integral geometry and geometric probability*. Cambridge: Cambridge University Press, 2004.
- [166] D. A. Klain and G.-C. Rota, *Introduction to geometric probability*. Cambridge: Cambridge University Press, 1997.
- [167] R. Schneider, “Kinematic measures for sets of colliding convex bodies,” *Mathematika*, vol. 25, no. 01, pp. 1–12, 1978.
- [168] R. Schneider and W. Weil, *Stochastic and integral geometry*. New York: Springer, 2008.
- [169] R. Schneider, “Integral geometric tools for stochastic geometry,” *Lecture Notes in Mathematics-Springer-Verlag*, vol. 1892, p. 119, 2007.

BIBLIOGRAPHY

- [170] G. Zhang, “A sufficient condition for one convex body containing another,” *Chinese Ann. Math. Ser. B*, vol. 9, no. 4, pp. 447–451, 1988.
- [171] J. Zhou, “A kinematic formula and analogues of Hadwiger’s theorem in space,” *Contemporary Mathematics*, vol. 140, pp. 159–159, 1992.
- [172] J. Zhou, “When can one domain enclose another in R^3 ?” *Journal of the Australian Mathematical Society-Series A*, vol. 59, no. 2, pp. 266–272, 1995.
- [173] J. Zhou, “Sufficient conditions for one domain to contain another in a space of constant curvature,” *Proceedings of the American Mathematical Society*, vol. 126, no. 9, pp. 2797–2803, 1998.
- [174] G. Boothroyd and A. H. Redford, *Mechanized Assembly: Fundamentals of parts feeding, orientation, and mechanized assembly*. New York: McGraw-Hill, 1968.
- [175] G. S. Chirikjian, “Parts entropy and the principal kinematic formula,” in *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on*, pp. 864–869, IEEE, 2008.
- [176] M. Karnik, S. K. Gupta, and E. B. Magrab, “Geometric algorithms for containment analysis of rotational parts,” *Computer-Aided Design*, vol. 37, no. 2, pp. 213–230, 2005.
- [177] S. Glasauer, “A generalization of intersection formulae of integral geometry,” *Geometriae Dedicata*, vol. 68, no. 1, pp. 101–121, 1997.

BIBLIOGRAPHY

- [178] S. Glasauer, “Translative and kinematic integral formulae concerning the convex hull operation,” *Mathematische Zeitschrift*, vol. 229, no. 3, pp. 493–518, 1998.
- [179] P. Goodey and W. Weil, “Translative integral formulae for convex bodies,” *aequationes mathematicae*, vol. 34, no. 1, pp. 64–77, 1987.
- [180] P. Goodey and W. Well, “Intersection bodies and ellipsoids,” *Mathematika*, vol. 42, pp. 295–304, 1995.
- [181] H. Groemer, “On translative integral geometry,” *Archiv der Mathematik*, vol. 29, no. 1, pp. 324–330, 1977.
- [182] R. Schneider and W. Weil, “Translative and kinematic integral formulae for curvature measures,” *Mathematische Nachrichten*, vol. 129, no. 1, pp. 67–80, 1986.
- [183] W. Weil, “Translative integral geometry,” *Geobild*, vol. 89, pp. 75–86, 1989.
- [184] W. Weil, “Translative and kinematic integral formulae for support functions,” *Geometriae Dedicata*, vol. 57, no. 1, pp. 91–103, 1995.
- [185] 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.
- [186] H. Ouadfel and L. Rothenburg, “An algorithm for detecting inter-ellipsoid contacts,” *Computers and Geotechnics*, vol. 24, pp. 245–263, June 1999.

BIBLIOGRAPHY

- [187] D. Henrion, S. Tarbouriech, and D. Arzelier, “LMI approximations for the radius of the intersection of ellipsoids: Survey,” *Journal of Optimization Theory and Applications*, vol. 108, no. 1, pp. 1–28, 2001.
- [188] S. Cameron, “A comparison of two fast algorithms for computing the distance between convex polyhedra,” *TR*, vol. 13, no. 6, pp. 915–920, 1997.
- [189] P. Jiménez, “3D collision detection: a survey,” *Computers & Graphics*, vol. 25, pp. 269–285, Apr. 2001.
- [190] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [191] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566–580, 1996.

Vita



Yan Yan was born in Xi'an, China in 1985. She obtained a B. S. in Automatic Control at Northwestern Polytechnic University, Xi'an, China in 2007 and earned her Master degree in Mechanical Engineering at Iowa State University in 2009. In 2009 she enrolled in the Mechanical Engineering Ph.D. program at Johns Hopkins University.

Yan Yan has been recognized with several distinctions including Honored Student Scholarship at Northwestern Polytechnic University from 2003-2007; Fellowship of Seward, Ratcliffe, and Galloway Foundation in Mechanical Engineering in 2008; Excellent Research Award at Iowa State University in 2009; Departmental Fellowship in Mechanical Engineering at Johns Hopkins University from 2009-2010; and American Control Conference Best Paper Award in 2010.