An Explicit Certified Method for Path Planning for an $SE(3)$ Robot

by

Zhaoqi Zhang

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

January, 2025

Chee Yap

© Zhaoqi Zhang

All Rights Reserved, 2025

Acknowledgements

I would like to thank my advisor, Chee Yap, for his full support and invaluable help with my project. He conceived the SSS framework 10 years ago and has continuously developed the theory until my generation. He provided tremendous assistance in formalizing the boundary reduction methods and gave extensive feedback to improve my English writing. He also coined precise and effective terminologies for my new concepts, which greatly aided me in explaining my ideas.

I am also deeply grateful to all the professors on my Ph.D. defense committee, in addition to Professor Chee Yap, including Professors Yi-Jen Chiang, Jingjin Yu, Daniele Panozzo, Anirudh Sivaraman, Dan Halperin, and David Xianfeng Gu. Professor Chiang collaborated with Professor Yap and me on the Delta robot path-planning project. His insights into computer graphics led to the development of the principal neighbor data structure, which plays a key role in accelerating the soft subdivision search. Professor Panozzo introduced me to the libigl library, on which I based all my graphical work. Professor Gu provided me with deeper insights into computational geometry and offered opportunities to present the SSS framework to the academic community.

I also want to express my heartfelt gratitude to my family and friends who supported me throughout my Ph.D. journey. My partner, Y. L., accompanied me through countless days and nights in the U.S., offering unwavering emotional support. My undergraduate friend, Xiaohan Yan, who pursues a career in theoretical mathematics, inspired me with discussions on Poincaré's duality, which contributed to the understanding of channel existence in the SSS framework. Finally, I want to thank my parents for their unwavering support, which enabled me to complete my Ph.D. degree.

Abstract

The design and implementation of theoretically-sound robot motion planning algorithms is challenging, especially for robots with high degrees of freedom (DOF). This thesis presents an explicit, practical and certified path planner for a rigid spatial robot with 6 DOFs. The robot is a spatial triangle moving amidst polyhedral obstacles. Correct, complete and practical path planners for such a robot has never been achieved. It is widely recognized as a key challenge in robotics. We design such a planner by using the Soft Subdivision Search (SSS) framework, based on the twin foundations of ε-exactness and soft predicates. This SSS planner is a theoretical alternative to the standard exact algorithms, and provides much stronger guarantees than probabilistic or sampling algorithms.

In this thesis, we address technical challenges for the $SE(3)$ robot. First, we establish the foundational theory of SSS framework by proving a general form of the Fundamental Theorem of SSS. Second, we introduce a topologically correct data structure for non-Euclidean path planning in the $SE(3)$ space. Third, we analyze the distortion bound of the $SE(3)$ representation. Fourth, we design an approximate footprint and combine it with the highly efficient feature set technique which leads to its soft predicate. Finally, we explicitly design the geometric primitives to avoid using a general solver of a polynomial system. This allows a direct implementation. These contributions represent a robust, practical, and adaptable solution to robot motion planning.

Keywords— Algorithmic Motion Planning, Subdivision Methods, Resolution-Exact Algorithms, Soft Predicates, Spatial 6 DOF Robots, Soft Subdivision Search.

Table of Contents

List of Figures

List of Tables

List of Appendices

Chapter 1

Introduction to Path Planning and the SSS framework

1.1 Preface

The history of motion planning is more than 40 years [36, 12] and continues as an ever-growing research field to the present. The popularity of motion planners is rising with the growing accessibility of inexpensive commercial robots, including autonomous mobile cleaners and package-delivery drones. Theoretically, motion planning is a broad field with numerous factors to consider, such as dynamic constraints involving forces, velocity, etc., and kinematic constraints related to robot geometry. However, the core problem is to plan a path for a robot to move from start positions to target positions [25] while navigating restrictions, such as external obstacles and internal constraints [55].

In this thesis, we will focus on the kinematic constraints only, which is also named as "Piano Mover's Problem" [30]. We concentrate on path planning, which in its elemental form involves finding a collision-free path from a starting position to a goal position for the robot, given a known map of the environment. Numerous methodologies for path planning are extensively documented in the literature. Essentially, there are three main approaches: Exact, Sampling and Subdivision [65]. The research of this thesis will be based on the subdivision approach.

Exact path planning approaches have been studied since the 1980s [27], and is reduced to the problem of checking connectivity of semi-algebraic sets (e.g., [17]). The output of an exact path planner is either a robot motion, or a NO-PATH indicator if no path exists. Naively, exact computation means to implement each step of a mathematically correct algorithm exactly. This would imply (for path planning) that each step must be computed using algebraic number operations (arithmetic operations, root extraction, etc) such as the **Cylindrical Algebraic Decomposition** (CAD) [7]. Unfortunately, such approaches including CAD are impractical. Even in simpler cases, correct implementations are rare for two reasons: it requires exact algebraic number computation and has numerous degenerate conditions (even in the plane) that are hard to enumerate or detect, e.g. [18]. There is a framework called **Exact Geometric Computation** (EGC) [54] that deals with this issue. Ultimately, it is to decide the sign of a numerical quantity x, where x might represent the minimum distance from the path to obstacles and we need x to be positive. This problem can be reduced to the "Zero Problem", which is to decide if $x = 0$. It can be supported by "EGC Number types" such as [61] or [38]. But in the worst case, it is exponentially expensive [7, 50].

During the last 30 years, practical path planning algorithms have flourished using the Sampling Approach. A standard textbook [15, p.201] describes the dominance of Sampling Approach in this area: "PRM, EST, RRT, SRT, and their variants have changed the way path planning is performed for high-dimensional robots. They have also paved the way for the development of planners for problems beyond basic path planning." Typically, such algorithms use a **sampling function** to pick samples and check whether the samples are valid in the environment or not. They use **local planners** to connect nearby samples, which grow a graph or a tree respectively by different sampling methods. People in this area call the discrete graph constructed from the samples a **road map**. The path-finding is reduced to finding a path in this graph [48]. Such algorithms face many issues such as how many sample points to take (before declaring that there is NO-PATH), and even when a path is found, the edges in the graph (given by local planners) are not guaranteed to be valid [48].

The origins of the Subdivision Approach can be traced to the early days of algorithmic robotics

[13, 68]. The approach of this thesis aligns with this approach. However, a new theoretical foundation is required. The new foundation is based on the interval method [40] and the concept of ε-exactness [58]. The interval idea, where guarantees for numerical approximations are needed [41], is encoded in the concept of soft predicates [56]. The ε-exactness approach avoids the "Zero Problem" that afflicts all exact geometric algorithms. Both of the two concepts that are mentioned above have significant implications for computational geometry, given that all exact algorithms inherently encounter Zero Problems.

Based on the soft predicates and ε-exactness, the Soft Subdivision Search (SSS) framework for path planning was formulated as a general framework [59]. There is a series of papers that shows the implementability and practices of SSS planners [59, 58, 24, 66, 23]. They included planar fat robots [24] and complex robots [66], as well as spatial 5-DOF robots (rod and ring) [23]. The implementations on rod and ring robots are the first rigorous and complete planner for 5-DOF spatial robots. In each case, experimental results demonstrated that SSS planners meet or exceed the performance of leading sampling algorithms [23]. This outcome is unexpected, given the much stronger theoretical guarantees of SSS, including its capability to decide NO-PATH.

In this thesis, we will review the SSS framework and rigorously complete the proof of the Resolution-Exact SSS theorem in [59]. Then we will design a complete, rigorous and practical planner for an $SE(3) = \mathbb{R}^3 \times SO(3)$ robot. Such a planner is a well-known challenge of path planning. Like similar challenges in the past (rod for $SE(2)$ and in $\mathbb{R}^3 \times S^2$), we choose a simple $SE(3)$ robot to demonstrate the principles. The robot is a planar isosceles right triangle $\mathcal{AOB}$ in $\mathbb{R}^3$, a.k.a. **Delta robot**. Note that our Delta robot is not the industrial Delta robot with the four-bar linkage structure.

The structure of this paper will be as follows: in Chapter 1, we clarify the basic concepts and notions, review the literature on the kinematic motion planning problem, and state the SSS framework. In Chapter 2, we establish the SSS axiom system and provide a complete proof of resolution-exactness theorem. In Chapter 3, we establish a systematic method by differential geometry for computing the atlas constant in the SSS framework. In Chapter 4, we compute the Lipschitz constant of the Delta robot and design the approximate footprint for the Delta robot. In Chapter 5, we review the double loop of collision detections on Σ_2 sets and establish the Boundary Reduction Method as an alternative that fixes a weakness of the double loop. In Chapter 6, we demonstrate the subdivision scheme and design a product structure for deciding adjacencies of $SE(3)$ boxes. In Chapter 7, we analyze an accurate resolution constant for the Delta robot; and summarize the experimental results.

1.2 Basic Concepts in Motion Planning and Notations

The concept of configuration space was first introduced in [36]. Later work defines it variably, depending on the context of each paper. What we consider important is the placement of the robot on each configuration, which we formalize with a footprint map. We define the configuration space by the footprint map for different robots.

Suppose we have a Hausdorff space $\mathcal{X}$ and a metric space $\mathcal{Z}$. We define the set of all compact subsets of $\mathcal{Z}$ to be a **footprint space**, denoted by $\mathcal{C}(\mathcal{Z})$. Let the distance function on $\mathcal{Z}$ be $d_{\mathcal{Z}}$, a **Hausdorff distance** defined on the power set of metric space $\mathcal{Z}$, denoted by $d_{\mathcal{H}}$ [49, 10], is defined as

$$d_{\mathcal{H}}(A, B) = \max \left\{ \sup_{a \in A} \inf_{b \in B} d_{\mathcal{Z}}(a, b), \sup_{b \in B} \inf_{a \in A} d_{\mathcal{Z}}(b, a) \right\}.$$

As a result, the Hausdorff distance restricted to $\mathcal{C}(\mathcal{Z})$ forms a metric. A **footprint map** is a continuous map

$$\mathrm{Fp} : \mathcal{X} \to \mathcal{C}(\mathcal{Z}) \tag{1.1}$$

with topology on $\mathcal{C}(\mathcal{Z})$ induced by $d_{\mathcal{H}}$. The domain $\mathcal{X}$ is called the **configuration space**, denoted by $\mathcal{X} = \mathrm{Cspace}$. Each point in the configuration space is called a **configuration**. The metric space $\mathcal{Z}$ is called the **physical space**.

An example of a footprint map discussed in this thesis is the footprint map for the **Delta robot**. The robot is denoted by $R_0 = \triangle\mathcal{AOB}$. Suppose the canonical placement of R_0 is $\mathcal{A} = \mathbf{e}_1 = (1, 0, 0)$, $\mathcal{B} = \mathbf{e}_2 = (0, 1, 0)$ and $\mathcal{O} = \mathbf{0} = (0, 0, 0)$. The configuration space for R_0 is $SE(3) = \mathbb{R}^3 \times SO(3)$, where we represent points in $SE(3)$ by $(\mathbf{x}, \sigma)$ such that $\mathbf{x} \in \mathbb{R}^3$ is a vector and $\gamma \in SO(3)$ is a rotation transformation on $\mathbb{R}^3$. The footprint map for R_0 is

$$SE(3) \to \mathcal{C}(\mathbb{R}^3)$$

$$(\mathbf{x}, \gamma) \mapsto \{\mathbf{x} + s\gamma(\mathbf{e}_1) + t\gamma(\mathbf{e}_2) : s, t \in [0, 1], s + t \leq 1\},$$

see Figure 1.1.

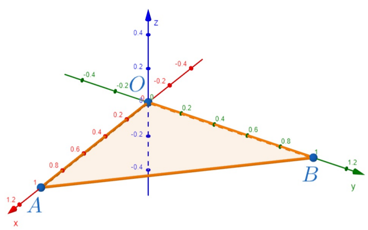


Figure 1.1: The canonical placement of the Delta robot.

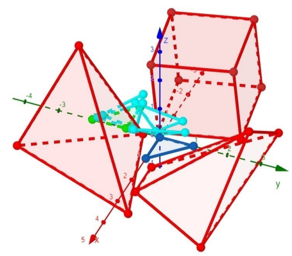


Figure 1.2: Footprints that form a path connecting two placements avoiding obstacles.

For any subset $S \subseteq$ Cspace, the **footprint** of the set, still denoted by Fp, is the union of footprints of all configurations in S, i.e.

$$\mathrm{Fp}(S) = \bigcup_{\gamma \in S} \mathrm{Fp}(\gamma).$$

The obstacle set Ω is a compact subset of the physical space $\mathcal{Z}$. We assume it to be a union of polyherals. In practice, it is represented by a set of features which forms a partition of $\partial\Omega$. The features consist of **corners** (points in $\mathbb{R}^3$), **edges** (line segments connecting corners), and **facets** (triangles formed by edges and corners). The feature set is denoted by Φ.

Given two subsets $A, B \subseteq \mathcal{Z}$, the **separation** between two sets is

$$\mathrm{Sep}(A, B) = \inf_{a \in A} \inf_{b \in B} d_{\mathcal{Z}}(a, b).$$

Suppose the obstacle set $\Omega \subseteq \mathcal{Z}$. For each configuration $\gamma \in$ Cspace, the clearance of the configuration is the separation between its footprint and the obstacle set, denoted by Cl, i.e.

$$\mathrm{Cl}(\gamma, \Omega) = \mathrm{Sep}(\mathrm{Fp}(\gamma), \Omega),$$

or simply written as $\mathrm{Cl}(\gamma)$ if Ω is fixed in the context. If $\mathrm{Cl}(\gamma) > 0$, then γ is called `FREE`, otherwise γ is called `STUCK`. The set of all `FREE` configurations is called the **free space**, denoted by $C_{free}(R_0, \Omega)$. Sometimes, we denote it by $\mathcal{Y}$, i.e.,

$$\mathcal{Y} = C_{free}(R_0, \Omega) = \{\gamma \in \mathcal{X} : \mathrm{Cl}(\gamma) > 0\}. \tag{1.2}$$

Similarly, the set of all `STUCK` configurations is called the **stuck space**, denoted by

$$C_{stuck}(R_0, \Omega). \tag{1.3}$$

By a **motion** of the robot R_0, we mean a continuous map $\pi : [0, 1] \to \mathrm{Cspace}$. We identify π with its image $\pi([0, 1])$. The **clearance** of the motion is the separation between its footprint and the obstacle set, or equivalently, the infimum of clearances of configurations in the motion, still denoted by Cl, i.e.

$$\mathrm{Cl}(\pi) = \inf_{\gamma \in \pi} \mathrm{Cl}(\gamma) = \mathrm{Sep}(\mathrm{Fp}(\pi), \Omega).$$

The motion is called an **Ω-avoiding path** (or simply, path) if $\pi(t) \in \mathcal{Y}$ for all $t \in [0, 1]$.

The (basic) path planning problem for R_0 is the following: suppose $\alpha, \beta \in \mathrm{Cspace}$, given (α, β, Ω), compute an Ω-avoiding path from α to β if one exists, otherwise return NO-PATH. An algorithm solving this problem is called a **path planner**.

A path planner is said to be **resolution-exact** (or ε-exact for short) if there exists some constant $K \geq 1$ that is independent to the input, such that for any input $(\alpha, \beta, \Omega, \varepsilon)$, the algorithm halts and satisfies the following two conditions:

- (P) if there is a path of clearance $K\varepsilon$, it returns a path;
- (N) if there is no path of clearance ε/K, it returns NO-PATH.

This was first proposed in [59]. Note that if the largest clearance is between ε/K and $K\varepsilon$, then we can output either a path or NO-PATH. This is a desirable and unique feature of the resolution-exact path planners.

At the end of this section, we list some notations and symbols used in this thesis. The origin of $\mathbb{R}^k$ is $\mathbf{0}$. The unit vector with the i-th component equal to 1 is $\mathbf{e}_i$. A ball with radius r and origin o is $\mathrm{Ball}(o, r)$. If $o = \mathbf{0}$, we simply denote it by $\mathrm{Ball}(r)$. A line segment connecting two points $\mathbf{p}$ and $\mathbf{q}$ is $\overline{\mathbf{pq}}$. A path in topology space T, $[0, 1] \to T$, is always denoted by the letter π. The volume (measure) of a set S is $\mathrm{Vol}(S)$. In this thesis, we will use bold font Latin alphabets like $\mathbf{p}$, $\mathbf{q}$ to denote vectors and points in $\mathbb{R}^k$ or $\mathcal{Z}$ and use greek letters like γ, ζ to denote points in $\mathcal{X}$ or $\mathcal{Y}$.

1.3 Review of Kinematic Motion Planning

LaValle [30] provides a comprehensive overview of path planning, while Halperin [22] offers a general survey of the topic. An early survey [60] describes two universal approaches to exact path planning: cell decomposition [51] and the retraction method [44, 47]. Since exact path planning is a semi-algebraic problem [52], it is reducible to general cylindrical algebraic decomposition techniques [6]. However, treating path planning as a connectivity problem results in single-exponential time complexity [50].

Planar rod, referred to as the "ladder", was first examined using cell decomposition in [51]. More efficient quadratic-time methods, based on the retraction approach, were introduced in [45, 46]. Spatial rods were initially addressed in [53]. The combinatorial complexity of its free space reaches $\Omega(n^4)$ in the worst case, and this is closely approximated by an $O(n^{4+\varepsilon})$ time algorithm [29]. Lee and Choset [32] present a planner for a 3D rod using the retraction method. Outside of SSS planners, a similar approach to Lee and Choset's work can be found in Nowakiewicz [43], who employs subdivision of the Cubic Model. However, like many subdivision techniques, this method ultimately samples configurations (at the corners or centers) within subdivision boxes and thus it is actually a sampling method. The results were highly favorable in comparison to pure sampling techniques (PRM). For sampling-based planners, the main task is determining whether a configuration is free, which is a well-known collision detection problem [35].

The sampling approach originated in 1979 [48] and has dominated the field since then. The history of sampling methods can be divided into four eras: the pre-sampling era, the sampling-advent era, the sampling-consolidation era, and the optimality and learning era [48]. Methods in the sampling approach, such as PRM and RRT (which we will refer to as PRM-like), used to face the halting problem. They can never confirm NO-PATH in sampling. The earliest work appearing to address the NO-PATH issue is by Zhu and Latombe [68, 67], whose "hierarchical framework" shares many features with the SSS framework. They made a nice observation that if there is no path in the adjacency graph of cells that are either `FREE` or `MIXED`, then it constitutes a proof of NO-PATH. But they still cannot detect NO-PATH. The non-termination issue persists.

Today, the sampling approach incorporates infeasibility proof techniques to construct a "wall" that blocks connectivity between start and goal configurations. An early reference for infeasibility proofs is Basch et al. [5], who aimed to provide proofs when a robot cannot pass through a "gate" in a 3D wall. Later, Zhang et al. [63] offered infeasibility proofs by establishing sufficient criteria for classifying a cell as `STUCK`. More recently, Li and Dantam [34] pursued infeasibility proofs through learning and other methods. Practitioners using sampling approaches combine PRM-like methods with infeasibility proofs, allowing both 'find path' and 'find wall' processes to run concurrently until one halts, returning a path or declaring NO-PATH. However, both processes require specific input guarantees. For PRM-like methods to find a path if one exists, the C_{free} must be ε-good[1] [28]. For infeasibility proofs to declare NO-PATH if no path exists, the C_{stuck} must be entirely ε-blocked[2] [33]. In general, it is not possible to verify whether C_{free} is ε-good or whether C_{stuck} is entirely ε-blocked. If the input does not satisfy both conditions, the processes may again fail to terminate.

[1] Given input (Cspace, Ω), there is a constant $\varepsilon > 0$ such that, for each point $p \in C_{free}$, the ratio between the volume of points connectable to p by local planners and the volume of Cspace is at least ε.

[2] Given input (Cspace, Ω), there is a constant $\varepsilon > 0$ such that, for each point $p \in C_{stuck}$, a closed ball of radius ε centered at p lies entirely within C_{stuck}.

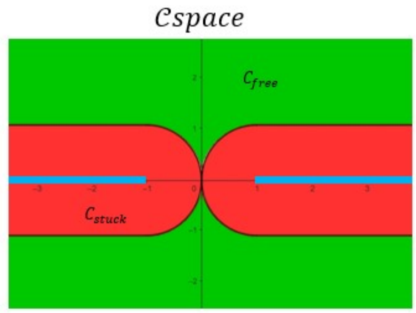


Figure 1.3: Square Cspace of disc robot with Ω be the two blue line segments. The green area represents C_{free}, and the red area represents C_{stuck}

Let us see Example 1.3.1 where a sampling approach combining PRM-like methods with infeasibility proof fails to halt.

Example 1.3.1. Let the footprint map Fp be defined as

$$\mathrm{Fp} : [-4, 4]^2 \to \mathcal{C}(\mathbb{R}^2)$$

$$\mathbf{x} \mapsto \{\mathbf{y} \in \mathbb{R}^2 : |\mathbf{x} - \mathbf{y}| \leq 1\},$$

representing a planar disc robot with radius 1. The obstacle is $\Omega = \{(x, y) \in \mathbb{R}^2 : |x| \geq 1, y = 0\}$. See Figure 1.3 for illustrations of C_{free} and C_{stuck}. Note that for any $\varepsilon > 0$, there exists a point $\mathbf{p} \in C_{free}$ such that, for $\mathbf{p}$ on y-axis and $|\mathbf{p}| < \min\{\frac{1}{3}\varepsilon, \frac{1}{4}\}$, the volume $\mathrm{Vol}(S(p))$, as described in [28], is bounded by $(4 - |\mathbf{p}|)^2 \tan(2 \arctan |\mathbf{p}|) < 48|\mathbf{p}| < 16\varepsilon$ (volume of Cspace is 16). Additionally, for any $\varepsilon > 0$, the point $\mathbf{0} \in C_{stuck}$ is not ε-blocked, as no ball $\mathrm{Ball}(\mathbf{0}, \varepsilon)$ can be entirely contained within C_{stuck}. Therefore, C_{free} is not ε-good, and C_{stuck} is not entirely ε-blocked. Thus, the sampling approach would never halt.

The theory of soft subdivision search is the first complete theory of path planning to resolve this halting issue in non-exact planners. The following series of papers demonstrate that this theory leads to implementable algorithms whose efficiency beats the state-of-the-art sampling methods, up to 5 DOF: [58, 56, 24, 66, 23]. This thesis extends these approaches to 6 DOF with $SE(3)$ configuration space, providing considerable efficiency for general cases and offering strong potential for real-time applications with appropriate heuristics. This represents the first implementation for such a class of robots that is complete, explicit, and resolves the halting issue.

1.4 On the SSS Framework

The SSS framework is based on the technique of soft predicate and classification on boxes. A **tile space** is $\mathcal{W} := \mathbb{R}^d$, where the SSS algorithm operates on tiles in $\mathcal{W}$, where $d \geq 1$ is at least the degree of freedom (DOF) of our robot. For $SE(3)$, $d = 7$ (not $d = 6$) because we immerse $SO(3)$ in R^4 to achieve the correct topology of $SO(3)$. We call a surjective local homeomorphism map $\mu : \mathcal{W} \to \mathcal{X}$ a **representation** of $\mathcal{X}$ by $\mathcal{W}$, which requires an atlas on $\mathcal{W}$ such that

1. μ restricted to each coordinate chart in the atlas is a homeomorphism;

2. μ^{-1} is an embedding from $\mathcal{X}$ to $\mathcal{W}$ restricted to each chart in the atlas.

Along the representation μ, each configuration $\gamma \in \mathcal{X}$ will be represented by one of the preimages in $\mu^{-1}(\gamma)$. We denote the domain of the representation by

$$B_0 = \mathrm{dom}(\mu). \tag{1.4}$$

The reason why we use a tile space to compute is that $\mathrm{dom}(\mu)$ can be decomposed into a union of tiles, where subdivision can be carried out using tiles. A **tile** is a d-dimensional, compact, and convex polytope of $\mathbb{R}^d$. For each tile B_0, a **subdivision** of B_0 is a finite set of tiles $\{B_1, \ldots, B_m\}$ such that $B_0 = \cup_{i=1}^m B_i$ and $\dim(B_i \cap B_j) < d$ for all $i \neq j$.

General tiles are beyond the present scope, so we restrict them to axes-parallel boxes and restrict $\mathcal{W}$ to **box space**. Let $\square\mathcal{W} = \square\mathbb{R}^d$ denote a set of boxes that may be used in a subdivision scheme. We define the footprint of each box $B \in \square\mathcal{W}$ by $\mathrm{Fp}(\mu(B))$ and simply denote it by $\mathrm{Fp}(B)$. In general, for simplicity, we identify points $\mathbf{b} \in \mathcal{W}$ with $\mu(\mathbf{b})$. So $\mathrm{Cl}(\mathbf{b})$ is also $\mathrm{Cl}(\mu(\mathbf{b}))$ for clearance function Cl.

Now we have terms for 4 space.

$$\mathcal{W} = \mathbb{R}^d, \quad \mathcal{X} = \text{Cspace}, \quad , \mathcal{Y} = C_{free}(R_0, \Omega), \quad \mathcal{Z} = \mathbb{R}^k. \tag{1.5}$$

Since $\mathcal{W}$ is a Euclidean space, we use bold font Latin alphabets to denote points also in $\mathcal{W}$ throughout the thesis.

Suppose `Expand` is a non-deterministic (i.e., multi-valued) function on $B \in \square\mathcal{W}$ such that `Expand`(B) is a subdivision of B. Using `Expand`, we can grow a subdivision tree $\mathcal{T}(B_0)$ rooted in $B_0 \in \square\mathcal{W}$ by repeatedly applying `Expand` to leaves of $\mathcal{T}(B_0)$. The set of leaves of $\mathcal{T}(B_0)$ forms a subdivision of B_0. Each of these boxes is represented by the product of intervals $B = \prod_{i=1}^{d} I_i$, where $I_i = [a_i, b_i] \subseteq \mathbb{R}$. For a box B, the **width** of B is $\mathrm{w}(B) = \min_{i=1}^{d} |I_i| = \min_{i=1}^{d} |b_i - a_i|$, and the **length** of B is $\ell(B) = \sup_{\mathbf{x},\mathbf{y} \in B} d_{\mathcal{W}}(\mathbf{x}, \mathbf{y})$. The **aspect ratio** is $\alpha(B) = \ell(B)/\mathrm{w}(B)$. Given a point $\mathbf{b} \in \mathcal{W}$ or a point $\gamma \in \mathcal{X}$ and a subdivision tree $\mathcal{T}(B_0)$, we write $\mathrm{Box}(\mathbf{b})$ as the unique leaf in the subdivision tree that contains $\mathbf{b}$ or $\mu^{-1}(\gamma)$.

Let Fp be the footprint map from Cspace to $\mathcal{C}(\mathcal{Z})$, Cl be the clearance function. An exact predicate C is a classification defined on each box $B \in \square\mathcal{W}$, which is a map from $\square\mathcal{W}$ to $\{\mathtt{FREE}, \mathtt{MIXED}, \mathtt{STUCK}\}$ by

$$C(B) = \begin{cases} \mathtt{FREE} & \forall \mathbf{b} \in B,\ \mathrm{Cl}(\mathbf{b}) > 0 \\ \mathtt{STUCK} & \forall \mathbf{b} \in B,\ \mathrm{Cl}(\mathbf{b}) = 0 \\ \mathtt{MIXED} & \text{otherwise} \end{cases}$$

To avoid exact computations, a soft predicate $\widetilde{C}$ is introduced. $\widetilde{C}$ is an approximation for C such that it is conservative and convergent, i.e.,

- (conservative) $\widetilde{C}(B) \neq \mathtt{MIXED}$ implies $\widetilde{C}(B) = C(B)$;
- (convergent) if $\lim_{i \to \infty} B_i = \{\mathbf{b}\}$, then there is $n \in \mathbb{N}$ such that when $i > n$, $\widetilde{C}(B_i) = C(\mathbf{b})$.

The SSS framework introduces a `MIXED` priority queue Q and a `FREE` graph G to proceed with searches. The priority queue Q is a collection of `MIXED`-leaves (by $\widetilde{C}$) with $\mathrm{w}(B) \geq \varepsilon$ that is waiting to be searched which is added from some `Expand`. The graph G is the collection of all `FREE` boxes (by $\widetilde{C}$) with w$(B) \geq \varepsilon$ where a Union-Find structure is maintained to trace if the boxes containing α and β (start and goal) are in the same connected component or not.

The general SSS framework process is the following:

SSS Framework

Input: Start configuration α, goal configuration β, obstacle Ω, resolution parameter ε.

Output: A path $\bar{P}$ or NO-PATH.

---

1. $\triangleright$ *Initialization*

   While $(\widetilde{C}(\text{Box}(\alpha)) \neq$ `FREE`),

      if w(Box(α)) $< \varepsilon$, return NO-PATH;

      else, `Expand`(Box(α)).

   While $(\widetilde{C}(\text{Box}(\beta)) \neq$ `FREE`),

      if w(Box(β)) $< \varepsilon$, return NO-PATH;

      else, `Expand`(Box(β)).

2. $\triangleright$ *Main Loop*

   While (`Find`(Box(α)) $\neq$ `Find`(Box(β))),

      if Q is empty, return NO-PATH

      $B \leftarrow Q$.GetNext()

      `Expand`(B).

3. $\triangleright$ *Search*

   Compute a `FREE` channel P from Box(α) to Box(β)

   Generate and return the canonical path $\overline{P}$ inside P.

The path $\overline{P}$ is defined as follows:

- Find the centers of the boxes in P;
- Find the centers of facets shared by adjacent boxes in P;
- Connecting the centers of boxes with their adjacent centers of facets by line segments;
- Return the union of the line segments as the path $\overline{P}$.

An important property for an SSS planner is that it is resolution-exact if the footprint map, the soft predicates and the subdivision schemes used in the planner satisfies 5 axioms. If there is a path π in $\mathcal{Y}$ with clearance higher than $K\varepsilon$, the SSS planner will construct a channel P. If there is no path π in C_{free} with essential clearance at least ε/K, the SSS planner will output NO-PATH. By a path with essential clearance ε/K, we mean a path $\pi : [0,1] \to \mathcal{Y}$, such that there is $a, b \in [0,1]$, such that $\mathrm{Sep}(\pi([a,b]), \Omega) > \varepsilon/K$. We define the term essential in order to avoid the possibility that α and β may be as close to Ω as possible. For example, given the input in Example 1.3.1 and any additional input resolution parameter ε, the SSS planner will return NO-PATH, since there is no path with essential clearance ε. This is the way that a SSS planner can avoid the halting problem. Compared to the sampling approach where PRM-like methods require C_{free} be ε-good and infeasibility proof requires C_{stuck} be entirely ε-blocked, the SSS planner detects resolution conditions by itself. The SSS framework provides an adaptive statement as output caters to different input.

In the next chapter, we will introduce the 5 axioms for SSS framework and prove that the SSS planner is resolution-exact by a Fundamental Theorem.

Chapter 2

SSS Axioms and Fundamental Theorem

Recall that the SSS framework consists of the following parts:

- We have a box space $\mathcal{W} = \mathbb{R}^d$, a configuration space $\mathcal{X} =$ Cspace, a free space $\mathcal{Y} = C_{free}$, and a physical space $\mathcal{Z} = \mathbb{R}^k$;
- There is a representation $\mu : \mathcal{W} \to \mathcal{X}$ which is a surjective local homeomorphism, and there is a continuous footprint map $\mathrm{Fp} : \mathcal{X} \to \mathcal{C}(\mathcal{Z})$;
- Subdivision of boxes takes place on box space $\mathcal{W}$, and the SSS path planner will find a channel P in the subdivision tree such that $\mu(P) \subseteq \mathcal{Y}$ or return NO-PATH.

The subdivision process begins by splitting an initial box B_0 to form a subdivision tree $\mathcal{T}(B_0)$. We denote the set of all leaves in a subdivision tree $\mathcal{T}(B_0)$ by $\mathcal{L}(\mathcal{T}(B_0))$.

In this chapter, we will discuss the 5 axioms of SSS framework with a simplified description, and prove the resolution-exactness of the SSS planner given the 5 axioms.

2.1 SSS Framework Formalization

Before talking about the axioms and the Fundamental theorem, we introduce the concepts of Lipschitz continuous, distortion, translational boxes and σ-effectiveness which are used in the axioms.

2.1.1 Lipschitz Continuous

We assume that the Cspace $\mathcal{X}$ is a metric space and denote the distance function of $\mathcal{X}$ by $d_{\mathcal{X}}$. The distance function on $\mathcal{C}(\mathcal{Z})$ is, by definition, the Hausdorff distance $d_{\mathcal{H}}$. Since both $\mathcal{X}$ and $\mathcal{C}(\mathcal{Z})$ are metric spaces, the continuity of Fp can be described by the usual ε-δ language, which is

$$\forall\gamma \in \mathcal{X},\ \forall\varepsilon > 0,\ \exists\delta > 0,\ \text{when } d_{\mathcal{X}}(\gamma,\zeta) < \delta,\ d_{\mathcal{H}}(\mathrm{Fp}(\gamma), \mathrm{Fp}(\zeta)) < \varepsilon.$$

Moreover, if there is a constant L_0, such that $\forall\gamma,\zeta \in \mathcal{X}$, we have

$$d_{\mathcal{H}}(\mathrm{Fp}(\gamma), \mathrm{Fp}(\zeta)) \leq L_0 d_{\mathcal{X}}(\gamma,\zeta),$$

we say that the footprint map Fp is **Lipschitz continuous**, and the constant L_0 is called a **Lipschitz constant**. A Lipschitz continuous Fp is uniformly continuous[1].

We can prove that the clearance Cl is a continuous function over $\mathcal{X}$. Let the obstacle $\Omega \subseteq \mathcal{Z}$ be a fixed closed set. Lemma A.1.1 in Section A.1 then implies that $\forall\gamma,\zeta \in \mathcal{X}$,

$$|\mathrm{Cl}(\gamma) - \mathrm{Cl}(\zeta)| = |\mathrm{Sep}(\mathrm{Fp}(\gamma), \Omega) - \mathrm{Sep}(\mathrm{Fp}(\zeta), \Omega)| \leq d_{\mathcal{H}}(\mathrm{Fp}(\gamma), \mathrm{Fp}(\zeta)).$$

The continuity of Fp proves that Cl is a continuous function.[2]

The free space $\mathcal{Y}$ is the subset of $\mathcal{X}$ where $\forall\gamma \in \mathcal{Y}$, $\mathrm{Cl}(\gamma) > 0$. Since the clearance function is continuous, $\mathcal{Y}$ is open in $\mathcal{X}$.

2.1.2 Representation Distortion

Representation map is a locally homeomorphism $\mu : \mathcal{W} \to \mathcal{X}$ such that its inverse is an embedding restricted to each chart in an atlas. Assume that both $\mathcal{W}$ and $\mathcal{X}$ are metric spaces. Given $\mathbf{p}, \mathbf{q} \in \mathcal{W}$ where $\gamma = \mu(\mathbf{p})$ and $\zeta = \mu(\mathbf{q})$, the **distortion** of μ between $\mathbf{p}$ and $\mathbf{q}$ is

$$\mathrm{tort}_{\mathbf{p},\mathbf{q}}(\mu) = \frac{d_{\mathcal{X}}(\gamma,\zeta)}{d_{\mathcal{W}}(\mathbf{p},\mathbf{q})}.$$

[1] $\forall\varepsilon > 0$, there is $\delta = \varepsilon/L_0$, such that when $d_{\mathcal{X}}(\gamma,\zeta) < \delta$, $d_{\mathcal{H}}(\mathrm{Fp}(\gamma), \mathrm{Fp}(\zeta)) \leq L_0 d_{\mathcal{X}}(\gamma,\zeta) < \varepsilon$.

[2] $\forall\varepsilon > 0$, there is $\delta > 0$, such that when $d_{\mathcal{X}}(\gamma,\zeta) < \delta$, $|\mathrm{Cl}(\gamma) - \mathrm{Cl}(\zeta)| \leq d_{\mathcal{H}}(\mathrm{Fp}(\gamma), \mathrm{Fp}(\zeta)) < \varepsilon$.

The range of this ratio for $\mathbf{p}, \mathbf{q} \in \mathcal{W}$ is the **distortion** of μ, denoted by $\mathrm{tort}(\mu)$. i.e.

$$\mathrm{tort}(\mu) = \left\{ \tau > 0 : \inf_{\mathbf{p},\mathbf{q} \in \mathcal{W}} \frac{d_{\mathcal{X}}(\gamma, \zeta)}{d_{\mathcal{W}}(\mathbf{p}, \mathbf{q})} \leq \tau \leq \sup_{\mathbf{p},\mathbf{q} \in \mathcal{W}} \frac{d_{\mathcal{X}}(\gamma, \zeta)}{d_{\mathcal{W}}(\mathbf{p}, \mathbf{q})} \right\}.$$

For simplicity, we define **lower bound** $m(\mu) = \inf_{\mathbf{p},\mathbf{q} \in \mathcal{W}} \frac{d_{\mathcal{X}}(\gamma, \zeta)}{d_{\mathcal{W}}(\mathbf{p}, \mathbf{q})}$ and **upper bound** $M(\mu) = \sup_{\mathbf{p},\mathbf{q} \in \mathcal{W}} \frac{d_{\mathcal{X}}(\gamma, \zeta)}{d_{\mathcal{W}}(\mathbf{p}, \mathbf{q})}$. Then, $\mathrm{tort}(\mu) = [m(\mu), M(\mu)]$.

A **distortion bound** is a number $C_0 \geq 1$ such that $\mathrm{tort}(\mu) \subseteq [\frac{1}{C_0}, C_0]$. If the distortion $\mathrm{tort}(\mu)$ has a bound C_0, then for all $\mathbf{p}, \mathbf{q} \in \mathcal{W}$, we have

$$\frac{1}{C_0} d_{\mathcal{W}}(\mathbf{p}, \mathbf{q}) < d_{\mathcal{X}}(\gamma, \zeta) < C_0 d_{\mathcal{W}}(\mathbf{p}, \mathbf{q}),$$

for $\gamma = \mu(\mathbf{p}), \zeta = \mu(\mathbf{q}) \in \mathcal{X}$.

2.1.3 Translational Boxes

"Translational" is a property of the footprint map that describes the translation motion of the robots. Suppose we have a purely translational robot R_0 whose canonical placement is $\mathcal{E}$. The footprint map for the robot R_0 is defined as:

$$\mathbb{R}^3 \to \mathcal{C}(\mathbb{R}^3)$$

$$\mathbf{x} \mapsto \{\mathbf{x}\} \oplus \mathcal{E},$$

where $\oplus$ is the Minkowski sum defined in Section A.4. If there is an obstacle set $\Omega \subseteq \mathbb{R}^3$, then we can explicitly describe $C_{stuck}(R_0, \Omega)$, which is

$$C_{stuck}(R_0, \Omega) = \Omega \oplus \mathcal{E}.$$

When both Ω and $\mathcal{E}$ are polyhedral sets, $C_{stuck}(R_0, \Omega)$ is also polyhedral. Then the complement $C_{free}(R_0, \Omega)$ can be easily decomposed into polyhedral cells [3], which is the foundation of many exact path planning algorithms.

The exact methods are hard to apply when rotation of robots are involved. The free space $C_{free}(R_0, \Omega)$ is no longer easily decomposable into cells. To deal with the problem, we exploit the translational property of $\mathcal{X}$, namely, it can be written as $\mathcal{X} = \mathcal{X}^t \times \mathcal{X}^r$, where the **translational space** $\mathcal{X}^t$ represents translations of the robot, and the **rotational space** $\mathcal{X}^r$ represents rotations of the robot. The two subspaces are combined by a Cartesian product. For the Delta robot $\mathcal{AOB}$, the configuration space $\mathcal{X} = SE(3) = \mathbb{R}^3 \times SO(3)$, where $\mathbb{R}^3$ corresponds to the translations and $SO(3)$ corresponds to the rotations.

A footprint map Fp is said to be **translational** if the domain $\mathcal{X} = \mathcal{X}^t \times \mathcal{X}^r$ such that $\mathcal{X}^t = \mathcal{Z}$ and for any closed subsets $\mathcal{I} \subseteq \mathcal{X}^t = \mathcal{Z}$, $\mathcal{K} \subseteq \mathcal{X}^r$,

$$\mathrm{Fp}(\mathcal{I} \times \mathcal{K}) = \mathcal{I} \oplus \mathrm{Fp}(\{\mathbf{0}\} \times \mathcal{K}). \tag{2.1}$$

In this case, Cspace $\mathcal{X}$ is called a translational configuration space. We say $\mathrm{Fp}(\{\mathbf{0}\} \times \{\mathrm{id}\})$ the **canonical placement** of the robot. Note that $\mathcal{X}^r$ can be the space with identity only. So the footprint maps of purely translational robots are translational. Moreover we can prove that the footprint map for the Delta robot is also translational:

Proposition 2.1.1. *The footprint map for the Delta robot is translational.*

Proof. For the Delta robot, $\mathcal{X} = \mathbb{R}^3 \times SO(3)$ where $\mathbb{R}^3 = \mathcal{Z}$ and the footprint map for any $\mathcal{I} \subseteq \mathbb{R}^3$ and $\mathcal{K} \subseteq SO(3)$ satisfies,

$$\begin{aligned}
\mathrm{Fp}(\mathcal{I} \times \mathcal{K}) &= \{\mathbf{x} + s\sigma(\mathbf{e}_1) + t\sigma(\mathbf{e}_2) : \mathbf{x} \in \mathcal{I}, \sigma \in \mathcal{K}, s, t \in [0,1], s + t \leq 1\} \\
&= \mathcal{I} \oplus \{\mathbf{0} + s\sigma(\mathbf{e}_1) + t\sigma(\mathbf{e}_2) : \sigma \in \mathcal{K}, s, t \in [0,1], s + t \leq 1\} \\
&= \mathcal{I} \oplus \mathrm{Fp}(\{\mathbf{0}\} \times \mathcal{K}).
\end{aligned}$$

$\square$

In this thesis, for simplicity, we identify $\mathrm{Fp}(\{\mathbf{0}\} \times \mathcal{K})$ with $\mathrm{Fp}(\mathcal{K})$ for any rotational subsets $\mathcal{K}$. Then a translational footprint map is

$$\mathrm{Fp}(\mathcal{I} \times \mathcal{K}) = \mathcal{I} \oplus \mathrm{Fp}(\mathcal{K}).$$

In SSS framework, a **translational** box space is $\mathcal{W} = \mathcal{W}^t \times \mathcal{W}^r$ and for each box $B \in \square\mathcal{W} = \square\mathbb{R}^d$,

$B = B^t \times B^r$ for $B^t \in \square\mathcal{W}^t = \square\mathbb{R}^k$ and $B^r \in \square\mathcal{W}^r = \square\mathbb{R}^{d-k}$ and

$$\mathrm{Fp}(B^t \times B^r) = B^t \oplus \mathrm{Fp}(B^r). \tag{2.2}$$

Boxes with property in Equation 2.2 are called **translational** boxes. Given a translational footprint map defined in Equation 2.1, we can construct such $\mathcal{W}$ by requiring the representation μ to be **decomposable**. That is, there are representations $\mu^t : \mathcal{W}^t \to \mathcal{X}^t$ and $\mu^r : \mathcal{W}^r \to \mathcal{X}^r$ such that μ^t is the identity map for $\mathcal{W}^t = \mathcal{X}^t = \mathcal{Z} = \mathbb{R}^k$ and $\mu((b^t, b^r)) = (\mu^t(b^t), \mu^r(b^r))$ for any $b^t \in \mathcal{W}^t$ and $b^r \in \mathcal{W}^r$.

Example 2.1.2. Let $\mu : \mathbb{R}^4 \to \mathbb{R}^2 \times SO(2)$ be

$$(x, y, z, w) \mapsto (x, y, \arctan(w/z)).$$

Then μ is decomposable as $\mu : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2 \times SO(2)$ such that

$$\mu^t : \mathcal{W}^t = \mathbb{R}^2 \to \mathcal{X}^t = \mathbb{R}^2(x, y) \qquad \mapsto (x, y)$$

and

$$\mu^r : \mathcal{W}^r = \mathbb{R}^2 \to \mathcal{X}^r = SO(2)(z, w) \qquad \mapsto \arctan(w/z)$$

A translational footprint map and a decomposable representation gives us a simple correspondence from the box space to the physical space via the Minkowski sum. As a summary, we review the correspondences throughout the representation and footprint maps according to Figure 2.1. The box space $\mathcal{W}$ represents the configuration space $\mathcal{X}$ by the representation map μ. Moreover, the box space can be decomposed into $\mathcal{W} = \mathcal{W}^t \times \mathcal{W}^r$ where $\mathcal{W}^t$ and $\mathcal{W}^r$ represents $\mathcal{X}^t$ and $\mathcal{X}^r$ via μ^t and μ^r respectively. The footprint map Fp maps subsets in $\mathcal{X}$ into subsets in $\mathcal{Z}$. For each subsets $\mathcal{I} \times \mathcal{K} \subseteq \mathcal{X}$ such that $\mathcal{I} \subseteq \mathcal{X}^t$ and $\mathcal{K} \subseteq \mathcal{X}^r$, $\mathrm{Fp}(\mathcal{I} \times \mathcal{K}) = \mathcal{I} \oplus \mathrm{Fp}(\mathcal{K})$. Especially, for $B^t \in \square\mathcal{W}^t$ and $B^r \in \square\mathcal{W}^r$, $\mathrm{Fp}(\mu(B^t \times B^r)) = B^t \oplus \mathrm{Fp}(B^r)$.

2.1.4 Effectiveness of Predicates

Recall that the classification of boxes in the SSS framework are based on soft predicates, that is a map from $\square\mathcal{W}$ to $\{\mathtt{FREE}, \mathtt{MIXED}, \mathtt{STUCK}\}$ such that it is conservative and convergent. For resolution-exactness,

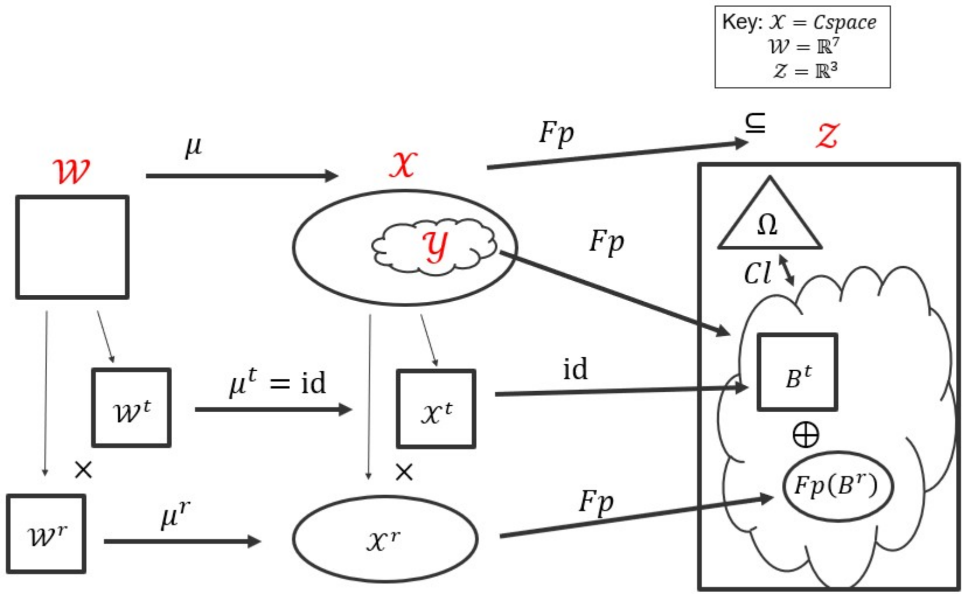


Figure 2.1: Translational maps in SSS framework.

we need another property: a soft predicate $\widetilde{C}$ is **effective** if there is an effectivity factor $\sigma \geq 1$ such that for each box $B \in \square\mathcal{W}$, if the exact predicate $C(\sigma B) = \texttt{FREE}$, then $\widetilde{C}(B) = \texttt{FREE}$. Such an effective soft predicate is called **σ-effective** [59]. The conservative of $\widetilde{C}$ and its σ-effectiveness results in the two direction predicate:

- If $\widetilde{C}(B) = \texttt{FREE}$, then $C(B) = \texttt{FREE}$;
- If $C(\sigma B) = \texttt{FREE}$, then $\widetilde{C}(B) = \texttt{FREE}$.

Note that an exact predicate is always an σ-effective soft predicate with $\sigma = 1$.

There is a general method to design σ-effective soft predicates which exploits methods of features [56]. Recall that the input of Ω is given by the primitive features. For each $B \in \square\mathcal{W}$, the **exact feature set** of B is

$$\phi(B) := \{f \in \Phi(\Omega) : f \cap \mathrm{Fp}(B) \neq \emptyset\}.$$

The exact predicate is given by the exact feature set as follow:

$$C(B) = \begin{cases} \texttt{FREE} & \phi(B) = \emptyset,\ \exists \mathbf{b} \in \mathrm{Fp}(B), \mathbf{b} \notin \Omega \\ \texttt{STUCK} & \phi(B) = \emptyset,\ \exists \mathbf{b} \in \mathrm{Fp}(B), \mathbf{b} \in \Omega \\ \texttt{MIXED} & \phi(B) \neq \emptyset \end{cases}$$

However, the exact feature set is too hard to compute, and we only want to design a soft predicate. The soft predicate will base on a similar process with an **approximate feature set** $\widetilde{\phi}(B)$ with the properties $\phi(B) \subseteq \widetilde{\phi}(B)$ and, to make it σ-effective, $\widetilde{\phi}(B) \subseteq \phi(\sigma B)$. This approximate feature set, for implementation, is defined by an **approximate footprint** $\widetilde{\mathrm{Fp}}$, such that

$$\widetilde{\phi}(B) := \{f \in \Phi(\Omega) : f \cap \widetilde{\mathrm{Fp}}(B) \neq \emptyset\}.$$

As required by the conservative and σ-effective, the approximate footprint $\widetilde{\mathrm{Fp}}$ should satisfy:

$$\mathrm{Fp}(B) \subseteq \widetilde{\mathrm{Fp}}(B) \subseteq \mathrm{Fp}(\sigma B)$$

for each $B \in \square\mathcal{W}$ [56].

A quick example of approximate footprint is a purely translational square robot $\mathcal{D} = [-1, 1]^2$ whose footprint map is

$$\mathrm{Fp} : \mathbb{R}^2 \to \mathcal{C}(\mathbb{R}^2)$$

$$\mathbf{x} \mapsto \{\mathbf{x}\} \oplus \mathcal{D}$$

The box space $\mathcal{W}$ for robot $\mathcal{D}$ is still $\mathbb{R}^2$ where the exact footprint for each box $B = [a, a+h] \times [b, b+h] \in \square\mathcal{W}$ is $\mathrm{Fp}(B) = [a-1, a+h+1] \times [b-1, b+h+1]$. The approximate footprint for each B can be defined as

$$\widetilde{\mathrm{Fp}}(B) = \mathrm{Ball}((a + h/2, b + h/2), \sqrt{2}(h + 2)/2),$$

which is the circumscribed ball of $\mathrm{Fp}(B)$. It is immediately that

$$\mathrm{Fp}(B) \subseteq \widetilde{\mathrm{Fp}}(B) \subseteq \mathrm{Fp}(\sqrt{2}B).$$

And hence, the soft predicate based on this approximate footprint is $\sqrt{2}$-effective.

The effectiveness gives adequate conditions for classification of boxes. See Section A.5 for more details.

2.2 SSS Axioms and the Fundamental Theorem

We have introduced the formalization of some important concepts in the SSS framework. More general concepts are introduced in Appendix A including the concept of **aspect ratio**. See Section A.3 for more details.

2.2.1 SSS Axioms

The resolution exactness of the SSS framework is based on the 5 axioms. They are:

Axiom 2.2.1 (**(A0)** Soft predicate.)**.** The soft predicate $\widetilde{C}$ is σ-effective.

Axiom 2.2.2 (**(A1)** Bounded aspect ratio.)**.** Cspace is a manifold and function `Expand`(B) splits each box B into at most finite subboxes. Moreover, there is a constant $D_0 \geq 1$ such that for all boxes $B \in \square\mathcal{W}$, the aspect ratio $\alpha(B) \leq D_0$, i.e., $l(B) \leq D_0 w(B)$.

Axiom 2.2.3 (**(A2)** Lipschitz footprint.)**.** The footprint map Fp is Lipschitz continuous, i.e., there is a constant $L_0 > 0$ such that for all $\gamma, \zeta \in \mathcal{X}$,

$$d_{\mathcal{H}}(\mathrm{Fp}(\gamma), \mathrm{Fp}(\zeta)) \leq L_0 d_{\mathcal{X}}(\gamma, \zeta).$$

Axiom 2.2.4 (**(A3)** Good atlas.)**.** The distortion of representation $\mathrm{tort}(\mu)$ is bounded, i.e., there is a constant $C_0 > 1$ such that $\forall \mathbf{p}, \mathbf{q} \in \mathcal{W}$ with $\gamma = \mu(\mathbf{p}), \zeta = \mu(\mathbf{q}) \in \mathcal{X}$,

$$\frac{1}{C_0} d_{\mathcal{W}}(\mathbf{p}, \mathbf{q}) < d_{\mathcal{X}}(\gamma, \zeta) < C_0 d_{\mathcal{W}}(\mathbf{p}, \mathbf{q}).$$

The constant C_0 is called **atlas constant**.

Axiom 2.2.5 (**(A4)** Translational box.)**.** Footprint map Fp is translational, representation μ is decomposable, and each box $B \in \square B_0$ is translational.

The halting problem of the SSS framework has been fully answered in [59]. In this section, we will prove that an SSS planner satisfying the 5 axioms is a resolution-exact planner. This result is called the **Fundamental Theorem** of SSS.

2.2.2 Fundamental Theorem of SSS

Recall that a resolution-exact path planner

- (P) returns a path if there is a path of clearance $K\varepsilon$;
- (N) returns NO-PATH if there is no path of clearance ε/K.

The **Fundamental Theorem** is the following:

Theorem 2.2.6 (Effective Fundamental Theorem)**.** *An SSS planner satisfying **(A0)**,**(A1)**,**(A2)**,**(A3)** and **(A4)** axioms is resolution-exact with an exact constant* $K = \max\{L_0 C_0 D_0 \sigma, 4D_0^2 L_0 C_0, 4\}$.

An SSS planner with Axiom **(A0)** sufficiently guarantees a correct path with positive clearance. The Lemma A.5.2 provides the correctness:

Lemma 2.2.7 (Correctness)**.** *If an SSS planner satisfies Axiom **(A0)**, the canonical path $\overline{P}$ given by the SSS planner has positive clearance.*

Proof. By the construction of $\overline{P}$, for each $\mathbf{b} \in \overline{P}$, $\mathbf{b}$ is in a `FREE` box B that is contained in the channel P. By Lemma A.5.2, $\mathrm{Cl}(\mathbf{b}) > 0$. Recall that Cl is continuous. For each point $\mathbf{b} \in \overline{P}$, there is an open neighborhood $\mathcal{U}_{\mathbf{b}}$ of $\mathbf{b}$, such that $\mathrm{Cl}(\mathcal{U}_{\mathbf{b}} \cap \overline{P}) \geq \frac{1}{2}\mathrm{Cl}(\mathbf{b}) > 0$. Then, since $\overline{P}$ is a finite closed set, by the Heine–Borel theorem [20], we can find a finite cover of $\overline{P}$ from $\cup_{\mathbf{b} \in \overline{P}} \mathcal{U}_{\mathbf{b}}$. i.e. $\overline{P} \subseteq \cup_{i=1}^{n} \mathcal{U}_{\mathbf{b}_i}$ for some $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \overline{P}$. Therefore,

$$\mathrm{Cl}(\overline{P}) \geq \min_{i=1}^{n} \mathrm{Cl}\left(\mathcal{U}_{\mathbf{b}_i} \cap \overline{P}\right) \geq \frac{1}{2} \min_{i=1}^{n} \mathrm{Cl}(\mathbf{b}_i) > 0.$$

Hence, $\overline{P}$ has positive clearance. $\square$

The Axiom **(A1)** declares the Cspace to be a manifold so that the root of the subdivision tree B_0 is restricted to a compact manifold. The manifold structure guarantees an important connectivity according to the Poincaré's duality. See more details in Section A.2 in Appendix A. As a simpler summary, we use Lemma 2.2.8 in this section.

Lemma 2.2.8 (Channel Lemma)**.** *If SSS planner satisfies axiom **(A1)**, then for each path $\pi : [0,1] \to \mathcal{W}$ and any subdivision tree $\mathcal{T}(B_0)$, there is a channel $P \subseteq \mathcal{L}(\mathcal{T}(B_0))$ that **exactly covers** π, i.e.*

i. $\pi(0) \in B_1$, $\pi(1) \in B_n$;

ii. for each box $B_i \in P$, there is $t \in [0,1]$ such that $\pi(t) \in B_i$.

Proof. Proposition A.2.2 shows $\texttt{Cover}(\pi)$ is a connected compact boxed homology manifold. Corollary A.2.4 constructs a channel connecting any pair of boxes in $\texttt{Cover}(\pi)$. We pick B_1 to be a box containing $\pi(0)$ and B_n to be a box containing $\pi(1)$ and construct a channel connecting B_1 and B_n to be the channel P. This channel P satisfies all conditions in the lemma. $\square$

2.2.3 Proof of Resolution Exactness

The proof of resolution exactness considers the worst case for an SSS planner to try to find a path, where all boxes in the subdivision tree that are not ε-small are splitted. i.e. All leaves of the subdivision tree are ε-small (with width between $\varepsilon/2$ and ε). We call this special subdivision tree an **ε-uniform** subdivision tree. We briefly explain the logic of the proof.

For the (P) part of resolution-exactness, if an SSS planner cannot find a path when there is a path π of clearance $K\varepsilon$, the subdivision tree when the planner halts will only consist of `FREE` boxes, `STUCK` boxes and ε-small boxes. Splitting each `FREE` or `STUCK` box will not change connectivity of the `FREE` zone, since no more distinct classifications will be formed. It is not a matter to keep subdividing the tree until it is uniformly ε. We can construct a channel in the ε-uniform subdivision tree that exactly covers π and prove that the channel is `FREE` by the axioms. Then the resulting channel contradicts the assumption that the SSS planner cannot find a path, which proves the (P) part.

For the (N) part of resolution-exactness, if an SSS planner returns a path when there is no path π with essential clearance ε/K, we compute the essential clearance of $\overline{P}$ and find it to be at least ε/K. The contradiction proves the (N) part.

We formulate the arguments of the two parts by two Lemmas 2.2.9 and 2.2.10.

Lemma 2.2.9 (Effective SSS)**.** *If the SSS planner satisfies axiom **(A0)**, **(A1)**, **(A2)** and **(A3)**, then the SSS planner can find a path $\overline{P}$ in a* `FREE` *channel P, if there exists a path $\pi \subseteq \mathcal{W}$ with clearance $L_0 C_0 D_0 \sigma\varepsilon$, where P exactly covers π.*

Proof. Suppose we have the uniformly ε subdivision tree $\mathcal{T}(B_0)$. Then there is a channel P that exactly covers π in $\mathcal{L}(\mathcal{T}(B_0))$ by Lemma 2.2.8. For each box $B \in P$, we compute the clearance of σB. Since P exactly covers π, there is $t \in [0,1]$ such that $\pi(t) \in B \subseteq \sigma B$. For each $\mathbf{b} \in \sigma B$,

$$\begin{aligned}
|\mathrm{Cl}(\mathbf{b}) - \mathrm{Cl}(\pi(t))| &\le d_{\mathcal{H}}(\mathrm{Fp}(\mathbf{b}), \mathrm{Fp}(\pi(t))) && \text{(Lemma } A.1.1) \\
&< L_0 d_{\mathcal{X}}(\mu(\mathbf{b}), \mu(\pi(t))) && \textbf{(A2)} \\
&< L_0 C_0 d_{\mathcal{W}}(\mathbf{b}, \pi(t)). && \textbf{(A3)} \\
&\le L_0 C_0 \ell(\sigma B) && \text{(box structure)} \\
&\le L_0 C_0 D_0 \mathrm{w}(\sigma B) && \textbf{(A1)} \\
&< L_0 C_0 D_0 \sigma\varepsilon. && \text{(Proposition } A.3.1)
\end{aligned}$$

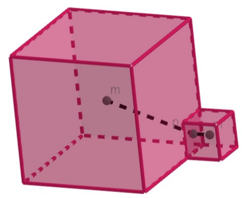
So $\forall \mathbf{b} \in \sigma B$,

$$\mathrm{Cl}(\mathbf{b}) > \mathrm{Cl}(\pi(t)) - L_0 C_0 D_0 \sigma\varepsilon \ge 0.$$

By Lemma A.5.1, each B in P is `FREE` in the SSS planner. So P will be found as a channel in the graph G of the SSS planner, and the planner will return a canonical path $\overline{P}$ from P. $\square$

Lemma 2.2.10 (Translational)**.** *If the SSS planner satisfies **(A0)**, **(A1)**, **(A2)** and **(A4)**, then the canonical path $\overline{P}$ given by the SSS planner has essential clearance $\frac{\varepsilon}{K}$ for $K = \max\{4D_0^2 L_0 C_0, 4\}$.*

Proof. Let $\overline{P}$ be the canonical path given by the SSS planner. The obstacle set is Ω. Recall that $\overline{P}$ is constructed by consecutive line segments that connect the centers of boxes with centers of facets that are shared by their adjacent boxes alternatively from channel P. Suppose that $\mathcal{W} = \mathcal{W}^t \times \mathcal{W}^r$, μ is decomposed into μ^t and μ^r, $B = B^t \times B^r$ is any box in P and $B' = B'^t \times B'^r$ is one of its adjacent boxes, $F = F^t \times F^r$ is the shared facet between B and B'. Note that B and B' may be different sized, and F is always a facet of the smaller box between B and B'. For $\mathrm{m}(B) = \mathbf{m} = (m^t, m^r)$ and $\mathrm{m}(F) = \mathbf{p} = (p^t, p^r)$, let's estimate $\mathrm{Cl}(\overline{\mathbf{mp}})$.



Since P collects boxes classified as `FREE` by $\widetilde{C}$, by Lemma A.5.2, $\mathrm{Cl}(B) > 0$ and hence $\mathrm{Sep}(\mathrm{Fp}(B), \Omega) >$ 0.

Then by **(A4)**, $\mathrm{Sep}(B^t \oplus \mathrm{Fp}(B^r), \Omega) = \mathrm{Sep}(\mathrm{Fp}(B), \Omega) > 0$, which implies

$$\begin{aligned}
\forall \mathbf{b} = (b^t, b^r) \in \overline{\mathbf{mp}},\ Cl(\mathbf{b}) &= \mathrm{Sep}(\mathrm{Fp}(\mathbf{b}), \Omega) && \text{(definition)} \\
&\geq \mathrm{Sep}(\mathrm{Fp}(\{b^t\} \times B^r), \Omega) && \text{(definition)} \\
&= \mathrm{Sep}(\{b^t\} \oplus \mathrm{Fp}(B^r), \Omega) && (\textbf{(A4)}) \\
&\geq \mathrm{Sep}(\{b^t\}, \partial B^t), && \text{(Lemma } A.4.2)
\end{aligned}$$

Similarly, since both B and B' are `FREE`, F is also `FREE` and hence $\mathrm{Cl}(F) > 0$. Then

$$\begin{aligned}
\mathrm{Cl}(\mathbf{p}) &= \mathrm{Sep}(\mathrm{Fp}(\mathbf{p}), \Omega) && \text{(definition)} \\
&\geq \mathrm{Sep}(\mathrm{Fp}(\{p^t\} \times F^r), \Omega) && \text{(definition)} \\
&\geq \mathrm{Sep}(\{p^t\} \oplus \mathrm{Fp}(F^r), \Omega) && (\textbf{(A4)}) \\
&\geq \mathrm{Sep}(\{p^t\}, \partial F^t) && \text{(Lemma } A.4.2)
\end{aligned}$$

$$\geq \frac{1}{2}\varepsilon. \qquad \text{(box structure)}$$

Now, for all $\mathbf{b} = (b^t, b^r) \in \overline{\mathbf{mp}}$, if $d_{\mathcal{W}}(\mathbf{b}, \mathbf{p}) < \frac{\varepsilon}{4L_0C_0}$,

$$\begin{aligned}
|\mathrm{Cl}(\mathbf{b}) - \mathrm{Cl}(\mathbf{p})| &\leq d_{\mathcal{H}}(\mathrm{Fp}(\mathbf{b}), \mathrm{Fp}(\mathbf{p})) && \text{(Lemma } A.1.1)\\
&\leq L_0 d_{\mathcal{X}}(\mu(\mathbf{b}), \mu(\mathbf{p})) && ((\mathbf{A2}))\\
&< L_0 C_0 d_{\mathcal{W}}(\mathbf{b}, \mathbf{p}) && ((\mathbf{A3}))\\
&< \frac{\varepsilon}{4},
\end{aligned}$$

and $\mathrm{Cl}(\mathbf{b}) \geq \mathrm{Cl}(\mathbf{p}) - \frac{\varepsilon}{4} \geq \frac{\varepsilon}{4}$; otherwise,

$$\begin{aligned}
\mathrm{Cl}(\mathbf{b}) &\geq \mathrm{Sep}(\{b^t\}, \partial B^t) && \text{(previous result)}\\
&\geq \frac{1}{\alpha(B)} d_{\mathcal{W}^t}(b^t, p^t) && \text{(Corollary } A.3.3)\\
&\geq \frac{1}{\alpha(B)} \frac{1}{D_0} d_{\mathcal{W}}(\mathbf{b}, \mathbf{p}) && ((\mathbf{A1}))\\
&\geq \frac{1}{D_0^2} d_{\mathcal{W}}(\mathbf{b}, \mathbf{p}) && ((\mathbf{A1}))\\
&= \frac{\varepsilon}{4D_0^2 L_0 C_0}.
\end{aligned}$$

Therefore,

$$\mathrm{Cl}(\overline{\mathbf{mp}}) = \inf_{\mathbf{b} \in \overline{\mathbf{mp}}} \mathrm{Cl}(\mathbf{b}) \geq \frac{\varepsilon}{\max\{4D_0^2 L_0 C_0, 4\}}.$$

Let $K = \max\{4D_0^2 L_0 C_0, 4\}$. Each of the line segments connecting centers of boxes with centers of faces has minimum clearance $\frac{\varepsilon}{K}$, so without the line segments connecting start and goal configurations, the union of those line segments also has minimum clearance $\frac{\varepsilon}{K}$. Therefore $\overline{P}$ has essential clearance $\frac{\varepsilon}{K}$ $\square$

Now we supplement the proof of the Fundamental Theorem.

Proof of the Fundamental Theorem 2.2.6. If there is a path of clearance $L_0 C_0 D_0 \sigma \varepsilon$, then by Lemma 2.2.9, the SSS planner can find a path $\overline{P}$ in a `FREE` channel P. Lemma 2.2.7 shows the correctness of the path. The (P) part is proved.

If there is no path of clearance $\frac{\varepsilon}{\max\{4D_0^2L_0C_0,4\}}$ but SSS planner still finds a path, then by Lemma 2.2.10, the path found by the SSS planner contradicts to the assumption. The contradiction implies the SSS planner must not find a path. The (N) part is proved.

As the result, the resolution constant is $K = \max\{L_0C_0D_0\sigma, 4D_0^2L_0C_0, 4\}$. $\square$

In fact, we have a more accurate resolution constant for SSS planners using purely box subdivisions. See Section A.6.

Chapter 3

Distortion Bounds for Representations

Recall that the resolution-exactness of an SSS planner is given by the 5 axioms **(A0)-(A4)** and the resolution constant K is determined by constants σ, D_0, L_0 and C_0. The constant C_0 in Axiom **(A3)** requires a bounded distortion for the representation map μ, where the distortion of μ is

$$\mathrm{tort}(\mu) = [m(\mu), M(\mu)] = \left\{ \tau > 0 : \inf_{\mathbf{p},\mathbf{q} \in \mathcal{W}} \frac{d_{\mathcal{X}}(\gamma, \zeta)}{d_{\mathcal{W}}(\mathbf{p}, \mathbf{q})} \leq \tau \leq \sup_{\mathbf{p},\mathbf{q} \in \mathcal{W}} \frac{d_{\mathcal{X}}(\gamma, \zeta)}{d_{\mathcal{W}}(\mathbf{p}, \mathbf{q})} \right\}.$$

The minimum distortion bound $C_0 = \max\{\frac{1}{m}, M\}$.

Given an arbitrary Cspace $\mathcal{X}$, it is not obvious to inspire the smallest atlas constant C_0. Previous works computed atlas constants for Cspace $= \mathbb{R}^2 \times S^1$ [24] and Cspace $= \mathbb{R}^3 \times S^2$ [23] by direct geometry arguments. As a remark, there is an error in [23]. See Example B.1.1 in Section B.1 for more detail.

In this chapter, we apply a systematic method to compute the atlas constant for the Delta robot $\mathcal{AOB}$. This method is discussed in Section B.3. We apply it to a special class of representations and combine the result with a double covering map.

3.1 Representation of $SE(3)$ Configuration Space

The configuration space $SE(3)$ is a non-Euclidean 6-dimensional space that lives naturally in 7-dimensions. To compute a path by box subdivisions, we use a decomposable representation map μ as follow:

- For representation of the translational subspace $\mathcal{W}^t \to \mathcal{X}^t = \mathbb{R}^3$, μ^t is the identity map;

- For representation of the rotational subspace $\mathcal{W}^r \to \mathcal{X}^r = SO(3)$, μ^r is

$$\mu^r : \partial[-1,1]^4 \to S^3 \to SO(3)$$

$$(w,x,y,z) \mapsto (a,b,c,d) \mapsto \begin{pmatrix} 2(a^2+b^2)-1 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & 2(a^2+c^2)-1 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & 2(a^2+d^2)-1 \end{pmatrix},$$

  where $(a,b,c,d) = (w,x,y,z)/\sqrt{w^2+x^2+y^2+z^2}$. Note that this μ^r is actually a double covering map with $\mu^r(w,x,y,z) = \mu^r(-w,-x,-y,-z)$. We identify (w,x,y,z) with $(-w,-x,-y,-z)$ and such quotient space is denoted by $\widehat{SO}(3)$.

- The representation of $SE(3)$ is

$$\mu : \mathbb{R}^3 \times \widehat{SO}(3) \to \mathbb{R}^3 \times SO(3)$$

$$(\mathbf{x}, \gamma) \mapsto (\mathbf{x}, \mu^r(\gamma))$$

  The space $\mathbb{R}^3 \times \widehat{SO}(3)$ is denoted by $\widehat{SE}(3)$.

The $\widehat{SE}(3)$ representation was first know to [14]. See Figure 3.1 for a model of μ^r.

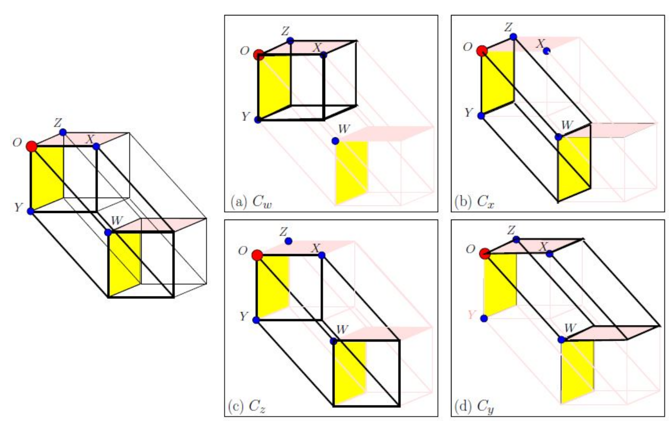


Figure 3.1: The representation $\mu^r : \widehat{SO}(3) \to SO(3)$.

3.2 On the Method of Distortion Bound

In this section, we discuss the distortion bound problem for a special class of representations, based on the Corollary B.3.2 in Section B.3. The idea of the distortion bound method turns the ratio of distance functions into ratios between tangent vectors. Let the Cspace $\mathcal{X}$ be S^n for any arbitrary $n \in \mathbb{N}^+$. There is a class of representations:

$$\mu_n : \partial[-1,1]^{n+1} \to S^n$$
$$\mathbf{x} \mapsto \frac{\mathbf{x}}{|\mathbf{x}|}$$

The distortion $\mathrm{tort}(\mu_n) = [\frac{1}{n+1}, 1]$, see Theorem B.4.1 in Section B.4. As examples, we compute the low-dimensional cases ($n = 2, 3$) explicitly for a correction of paper [23] and as a lemma for the distortion bounds for representation of $SE(3)$.

3.2.1 Distortion Bound for μ_2

The case for $n = 2$ is the Proposition 3.2.1:

Proposition 3.2.1. *The distortion of representation*

$$\mu_2 : \partial[-1,1]^3 \to S^2$$
$$(x, y, z) \mapsto \left(\frac{x}{r}, \frac{y}{r}, \frac{z}{r}\right),$$

where $r = \sqrt{x^2 + y^2 + z^2}$ *is* $[\frac{1}{3}, 1]$.

Proof. Since the 6 faces $B_1, \ldots, B_6$ are symmetric, we may use the face B_6 ($z = 1$) without loss of generality. For the point $\mathbf{p} = (x, y, 1) \in B_6$ where $r = \sqrt{x^2 + y^2 + 1}$, μ_2 maps it to the point

$$\gamma = (a, b, c) = \left(\frac{x}{r}, \frac{y}{r}, \frac{1}{r}\right).$$

For each $v_{\mathbf{p}} = \mathrm{dx}\frac{\partial}{\partial x} + \mathrm{dy}\frac{\partial}{\partial y} \in T_{\mathbf{p}}B_6$, the push forward map $(\mu_2)_*$ is given by

$$
\begin{aligned}
\mathrm{da}\frac{\partial}{\partial a} + \mathrm{db}\frac{\partial}{\partial b} + \mathrm{dc}\frac{\partial}{\partial c} &= (\mu_2)_*\left(\begin{pmatrix} \mathrm{dx} & \mathrm{dy} \end{pmatrix}\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix}\right) \\
&= \begin{pmatrix} \mathrm{dx} & \mathrm{dy} \end{pmatrix}(\mu_2)_*\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \qquad \text{(Push forward is linear)} \\
&= \begin{pmatrix} \mathrm{dx} & \mathrm{dy} \end{pmatrix} J_{\mu_2}\begin{pmatrix} \frac{\partial}{\partial a} \\ \frac{\partial}{\partial b} \\ \frac{\partial}{\partial c} \end{pmatrix}
\end{aligned}
$$

where J_{μ_2} is the Jacobian of μ_2 on B_6, i.e.,

$$
J_{\mu_2} = \begin{pmatrix} \frac{\partial a}{\partial x} & \frac{\partial b}{\partial x} & \frac{\partial c}{\partial x} \\ \frac{\partial a}{\partial y} & \frac{\partial b}{\partial y} & \frac{\partial c}{\partial y} \end{pmatrix}.
$$

For any $\gamma \in T_\gamma S^2$, $g_{S^2}\langle v_\gamma, v_\gamma\rangle$ is given by the Riemannian metric inherited from $\mathbb{R}^3$, which is

$$
g_{S^2}\langle v_\gamma, v_\gamma\rangle = (\mathrm{da}, \mathrm{db}, \mathrm{dc})(\mathrm{da}, \mathrm{db}, \mathrm{dc})^T.
$$

Therefore, for any $v_{\mathbf{p}} \in T_{\mathbf{p}}B_6$ such that $g_{B_6}\langle v_{\mathbf{p}}, v_{\mathbf{p}}\rangle = 1$, where g_{B_6} is also inherited from $\mathbb{R}^3$ which implies $1 = g_{B_6}\langle v_{\mathbf{p}}, v_{\mathbf{p}}\rangle = (\mathrm{dx}, \mathrm{dy})(\mathrm{dx}, \mathrm{dy})^T$,

$$
\begin{aligned}
\mu_2^*\langle v_{\mathbf{p}}, v_{\mathbf{p}}\rangle &= g_{S^2}\langle (\mu_2)_*(v_{\mathbf{p}}), (\mu_2)_*(v_{\mathbf{p}})\rangle \\
&= \begin{pmatrix} \mathrm{da} & \mathrm{db} & \mathrm{dc} \end{pmatrix}\begin{pmatrix} \mathrm{da} & \mathrm{db} & \mathrm{dc} \end{pmatrix}^T \\
&= \begin{pmatrix} \mathrm{dx} & \mathrm{dy} \end{pmatrix} J_{\mu_2} J_{\mu_2}^T \begin{pmatrix} \mathrm{dx} & \mathrm{dy} \end{pmatrix}^T \\
&= \frac{1}{(x^2+y^2+1)^2}\begin{pmatrix} \mathrm{dx} & \mathrm{dy} \end{pmatrix}\begin{pmatrix} y^2+1 & -xy \\ -xy & x^2+1 \end{pmatrix}\begin{pmatrix} \mathrm{dx} \\ \mathrm{dy} \end{pmatrix} \\
&= \frac{(x\mathrm{dx} - y\mathrm{dy})^2 + (\mathrm{dx}^2 + \mathrm{dy}^2)}{(x^2+y^2+1)^2}.
\end{aligned}
$$

The bounds for the last fraction can be estimated by the following two methods:

1.

$$\begin{aligned}
\frac{(x\mathrm{dx}-y\mathrm{dy})^2+(\mathrm{dx}^2+\mathrm{dy}^2)}{(x^2+y^2+1)^2} &\le \frac{(x^2+y^2)(\mathrm{dx}^2+\mathrm{dy}^2)+(\mathrm{dx}^2+\mathrm{dy}^2)}{(x^2+y^2+1)^2} && \text{(Cauchy's inequality)}\\
&= \frac{1}{x^2+y^2+1} && (\mathrm{dx}^2+\mathrm{dy}^2=1)\\
&\le 1.
\end{aligned}$$

where the maximum reaches when $x = y = 0$ for any $(\mathrm{dx}, \mathrm{dy}) \in \mathbb{R}^2$.

2.

$$\begin{aligned}
\frac{(x\mathrm{dx}-y\mathrm{dy})^2+(\mathrm{dx}^2+\mathrm{dy}^2)}{(x^2+y^2+1)^2} &= \frac{(x\mathrm{dx}-y\mathrm{dy})^2+1}{(x^2+y^2+1)^2} && (\mathrm{dx}^2+\mathrm{dy}^2=1)\\
&\ge \frac{1}{(x^2+y^2+1)^2} && ((x\mathrm{dx}-y\mathrm{dy})^2\ge 0)\\
&\ge \frac{1}{9}.
\end{aligned}$$

where the minimum reaches when $x = y = 1$ for $(\mathrm{dx}, \mathrm{dy}) = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$.

By Corollary B.3.2, $\mathrm{tort}(\mu_2) = [\frac{1}{3}, 1]$. $\square$

Corollary 3.2.2. *The atlas constant for an $\mathbb{R}^3 \times S^2$ robot (rod, ring ,etc.) by the representation $\mathbb{R}^3 \times \widehat{S}^2$ [23] is 3. But one can define $\mathcal{W} = \mathbb{R}^3 \times \partial[-\sqrt{3}^{-1}, \sqrt{3}^{-1}]^3$ to reduce the distortion bound down to $\sqrt{3}$.*

3.2.2 Distortion Bound for μ_3

The case for $n = 3$ is the Lemma 3.2.3:

Lemma 3.2.3. *The distortion of representation*

$$\begin{aligned}
\mu_3 : \partial[-1,1]^4 &\to S^3\\
(w,x,y,z) &\mapsto \left(\frac{w}{r}, \frac{x}{r}, \frac{y}{r}, \frac{z}{r}\right),
\end{aligned}$$

where $r = \sqrt{w^2+x^2+y^2+z^2}$ is $[\frac{1}{4}, 1]$.

Proof. $\partial[-1,1]^4$ is the union of 8 cubes with $w, x, y, z = \pm 1$ respectively on different cubes. We use the cube $B_1 = \{(w,x,y,z) \in \partial[-1,1]^4 : w = 1\}$ as an example. By the similar process in Proposition 3.2.1, for any $\mathbf{p} = (1,x,y,z) \in B_1$ and any $v_{\mathbf{p}} = \mathrm{dx}\frac{\partial}{\partial x} + \mathrm{dy}\frac{\partial}{\partial y} + \mathrm{dz}\frac{\partial}{\partial z} \in T_{\mathbf{p}}B_1$ such that $g_{B_1}\langle v_{\mathbf{p}}, v_{\mathbf{p}}\rangle = 1$,

$$\mu_3^*\langle v_{\mathbf{p}}, v_{\mathbf{p}}\rangle = \begin{pmatrix} \mathrm{dx} & \mathrm{dy} & \mathrm{dz} \end{pmatrix} J_{\mu_3} J_{\mu_3}^T \begin{pmatrix} \mathrm{dx} & \mathrm{dy} & \mathrm{dz} \end{pmatrix}^T$$
$$= \frac{1}{(1+x^2+y^2+z^2)^2} \begin{pmatrix} \mathrm{dx} & \mathrm{dy} & \mathrm{dz} \end{pmatrix} \begin{pmatrix} y^2+z^2+1 & -xy & -xz \\ -xy & x^2+z^2+1 & -yz \\ -xz & -yz & x^2+y^2+1 \end{pmatrix} \begin{pmatrix} \mathrm{dx} \\ \mathrm{dy} \\ \mathrm{dz} \end{pmatrix}$$

The bounds for the quadratic form can be estimated by the following two methods:

1.

$$\begin{aligned} \mu_3^*\langle v_{\mathbf{p}}, v_{\mathbf{p}}\rangle &= \frac{(1+x^2+y^2+z^2) - (x\mathrm{dx}+y\mathrm{dy}+z\mathrm{dz})^2}{(1+x^2+y^2+z^2)^2} && (\mathrm{dx}^2+\mathrm{dy}^2+\mathrm{dz}^2 = 1) \\ &\le \frac{1+x^2+y^2+z^2}{(1+x^2+y^2+z^2)^2} && ((x\mathrm{dx}+y\mathrm{dy}+z\mathrm{dz})^2 \ge 0) \\ &\le 1 && (1+x^2+y^2+z^2 \ge 1) \end{aligned}$$

where the maximum reaches when $x = y = z = 0$ for any $(\mathrm{dx}, \mathrm{dy}, \mathrm{dz}) \in \mathbb{R}^3$.

2.

$$\begin{aligned} \mu_3^*\langle v_{\mathbf{p}}, v_{\mathbf{p}}\rangle &= \frac{(x\mathrm{dy}-y\mathrm{dx})^2 + (x\mathrm{dz}-z\mathrm{dx})^2 + (y\mathrm{dz}-z\mathrm{dy})^2 + (\mathrm{dx}^2+\mathrm{dy}^2+\mathrm{dz}^2)}{(1+x^2+y^2+z^2)^2} \\ &= \frac{(x\mathrm{dy}-y\mathrm{dx})^2 + (x\mathrm{dz}-z\mathrm{dx})^2 + (y\mathrm{dz}-z\mathrm{dy})^2 + 1}{(1+x^2+y^2+z^2)^2} && (\mathrm{dx}^2+\mathrm{dy}^2+\mathrm{dz}^2 = 1) \\ &\ge \frac{1}{(1+x^2+y^2+z^2)^2} \\ &\ge \frac{1}{16}. \end{aligned}$$

where the minimum reaches when $x = y = z = 1$ for $(\mathrm{dx}, \mathrm{dy}, \mathrm{dz}) = \left(\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right)$.

By Corollary B.3.2, $\mathrm{tort}(\mu_3) = [\frac{1}{4}, 1]$. □

3.3 Atlas Constant for an $SE(3)$ Cspace

The $SE(3)$ rotational subspace $\mathcal{X}^r = SO(3)$ is not exactly the manifold S^3 in the Lemma 3.2.3. There is a double covering map from S^3 to $SO(3)$. The box subspace $\mathcal{W}^r = \widehat{SO}(3)$ is also a double covering map from $\partial[-1,1]^4$ but whose metric is inherited from the double covering. So the metric on $\widehat{SO}(3)$ is the same as the metric on $\partial[-1,1]^4$. Their relation can be given by the following commutative diagram, we will follow the symbols along the diagram throughout this section:

$$\begin{tikzcd}
\partial[-1,1]^4 \arrow[r, "\mu_3"] \arrow[d, "\mathbf{b}\sim(-\mathbf{b})"'] & S^3 \arrow[d, "\gamma\sim(-\gamma)"'] \arrow[dr, "\rho"] & \\
\widehat{SO}(3) \arrow[r, "\mu_3/\{\pm 1\}"'] & \mathbb{RP}^3 \arrow[r, "\rho/\{\pm 1\}"'] & SO(3)
\end{tikzcd}$$

The representation μ^r is the composition of ρ and μ_3. By Lemma 3.2.3, $\mathrm{tort}(\mu_3) = [\frac{1}{4}, 1]$. However, the distortion bound for ρ is indeterminate without an explicit metric on $SO(3)$. The choice of an essential metric on $SO(3)$ is a problem.

In Huynh's paper [26], 6 different metrics are defined on $SO(3)$ under different representations. In the paper [26], metric Φ_1 is the Euclidean distance between Euler angles $\|(\Delta\alpha, \Delta\beta, \Delta\gamma)\|_2$, Φ_2 is the norm of the difference of quaternions (S^3) $\min\{|q_1 - q_2|, |q_1 + q_2|\}$, Φ_3 is the angle between unit quaterions $\arccos|q_1 \cdot q_2|$, Φ_4 is one minus the absolute value of inner product of unit quaterions $1 - |q_1 \cdot q_2|$, Φ_5 is the Frobenius norm of identity matrix minus the transition matrix $\|I - R_1 R_2^T\|_F$, and Φ_6 is the distance by the exponential map $\|\log(R_1 R_2^T)\|$. Note that Φ_3 is the metric induced by ρ, i.e., $\Phi_3(q_1, q_2) = g_{S^3}(q_1, q_2)$, and we choose Φ_6 as the **essential metric** on $SO(3)$. This Φ_6 will be used to compute both atlas constant C_0 and Lipschitz constant L_0. In the paper [26], it is pointed out that $\Phi_6 = 2\Phi_3$, therefore $\mathrm{tort}(\rho) = \{2\}$ and the distortion for $\mu^r = \rho \circ \mu_3$ is $\mathrm{tort}(\mu^r) = [\frac{1}{2}, 2]$. To explain the metrics defined in [26] and the choice of essential metric, let's first review the space $SO(3)$.

3.3.1 Review of $SO(3)$

The Cspace $SO(3)$ consists of all positively defined orthogonal maps on $\mathbb{R}^3$. It is the set of all 3×3 matrices R such that $R^TR = I$. Let $R \in SO(3)$, for each $v \in \mathbb{R}^3$, $|Rv| = (Rv)^T(Rv) = v^TR^TRv = v^tv = |v|$. Therefore, the norm of all eigenvalues of R are 1. The characteristic polynomial $p_R(\lambda) = \det(R-\lambda I)$ is a real coefficient polynomial with degree 3 and hence there is a real root for $p_R(\lambda)$. This real root is $+1$ (not -1) since R is positively defined, otherwise for the eigenvector v corresponds to -1, $v^TRv = -v^Tv < 0$. Then the two image roots are $e^{i\theta}$ and $e^{-i\theta}$ for some $\theta \in S^1$. The unit eigenvector $v \in S^2$ corresponding to 1 is called the **axis** of the rotation, while the angle θ for the image roots is called the **angle** of the rotation [4] (the identity map has no axis or angle). The rotation matrix with axis $v \in S^2$ and angle $\theta \in S^1$ is denoted by $R(v, \theta)$. The **Lie Bracket** of an axis $v = (x, y, z)$ is a map $[\cdot] : \mathbb{R}^3 \to \mathfrak{so}(3)$, where $\mathfrak{so}(3)$ is the set of all 3×3 matrices V with $V + V^T = 0$, such that

$$[v] = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}.$$

The Lie bracket of an axis represents its cross product with other vectors, i.e., $\forall u, v \in \mathbb{R}^3$, $[v]u = v \times u$, and $[v]^3 = -|v|^2[v]$. The **exponential map** on $\theta[v]$ for axis v and angle θ returns the rotation matrix with the Euler-Rodrigues formula [37]:

$$R(v, \theta) = e^{\theta[v]} = I + \sin\theta[v] + (1 - \cos\theta)[v]^2.$$

The **logarithm** of a matrix $R(v, \theta) \in SO(3)$ is $\log(R(v, \theta)) = \theta[v]$ defined by this exponential map. For the metric Φ_6 in Huynh's paper [26], $R(v, \theta) = R_1R_2^T$ is the transition map from R_2 to R_1 since $(R_1R_2^T)R_2 = R_1(R_2^TR_2) = R_1$. Then $\Phi_6(R_1, R_2) = \|\log(R_1R_2^T)\| = \|\theta[v]\| = \theta$ is the angle of this transition map. This natural angle defines the essential metric on $SO(3)$.

The quaternion representaion of $SO(3)$ is a diffeomorphism from $\mathbb{RP}^3$ to $SO(3)$. The **unit quaternion** Q is the space S^3 endowed with a division ring structure, such that for each $r \in Q$, $r = a+b\mathbf{i}+c\mathbf{j}+d\mathbf{k}$

where $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$, $\mathbf{ij} = -\mathbf{ji} = \mathbf{k}$, $\mathbf{jk} = -\mathbf{kj} = \mathbf{i}$, $\mathbf{ki} = -\mathbf{ik} = \mathbf{j}$ [1]. The component w is called the **real** part of r while $b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ is called the **image** part of r. The **conjugate** of a unit quaternion $r = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ is $\overline{r} = a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}$. Obviously, $r\overline{r} = 1$ for any $r \in Q$, which implies $\overline{r} = r^{-1}$. A unit quaternion with real part $w = 0$ is called a **pure quaternion**. Such pure quaternions can represent vectors in $\mathbb{R}^3$ by identifying $b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ with (b, c, d), which form a subgroup by products in Q, denoted by $\mathbb{H}_p$ [19]. For any pure quaternion $v = v_i\mathbf{i} + v_j\mathbf{j} + v_k\mathbf{k} \in \mathbb{H}_p$ and any $\theta \in S^1$, $r = \cos\theta + v\sin\theta$ defines a unit quaternion. Moreover, such a unit quaternion defines a group action on $\mathbb{H}_p$ [19] by $r \circ u = rur^{-1}$ for any $r \in Q$ and any $u \in \mathbb{H}_p$. This group action on $\mathbb{H}_p$ is crucial since it defines the correspondence $\rho : S^3 = Q \to SO(3)$ by

$$\rho(r)(u) = rur^{-1}, \ \forall u \in \mathbb{H}_p = \mathbb{R}^3$$

It can be prove that for $r = \cos\theta + v\sin\theta$, $\rho(r) = R(v, 2\theta)$, see Theorem B.5.1 in Section B.5 [19]. Since r and $-r$ induce the same action on $\mathbb{H}_p$, ρ is a double covering map with $r \sim -r$. For the metric Φ_3 in Huynh's paper [26], if for $q_1, q_2 \in Q$, $r = \cos\theta + v\sin\theta = q_1q_2^{-1}$, $\theta \in [0, \pi]$, then

$$\begin{aligned}
\Phi_3(q_1, q_2) &= \arccos|q_1 \cdot q_2| \\
&= \arccos|(\cos\theta + v\sin\theta)q_2 \cdot q_2| \\
&= \arccos|\cos\theta q_2 \cdot q_2 + \sin\theta(vq_2) \cdot q_2| \\
&= \arccos|\cos\theta + 0| \\
&= \min\{\theta, \pi - \theta\},
\end{aligned}$$

where $(vq_2) \cdot q_2 = 0$ is a direct computation. Note that when $\Phi_3(q_1, q_2) = \pi - \theta$ ($\theta > \frac{\pi}{2}$), $2\Phi_3(q_1, q_2) = 2\Phi(q_1, -q_2) = 2\theta - 2\pi \cong 2\theta \in S^1$. Therefore, $\Phi_6(\rho(q_1), \rho(q_2)) = 2\theta = 2\Phi_3(q_1, q_2)$ for any pair of $q_1, q_2 \in Q$. It proves that $\Phi_6 = 2\Phi_3$.

By definition, the distortion for map $\rho : S^3 \to SO(3)$ is $\mathrm{tort}(\rho) = \{2\}$.

3.3.2 Distortion Bound for $SE(3)$

The map $\mu^r = \rho \circ \mu_3$ where the distortion of μ_3 is $[\frac{1}{4}, 1]$ and the distortion of ρ is $\{2\}$. Therefore, the distortion of μ^r is $[\frac{1}{2}, 2]$.

The Cspace $\mathcal{X} = SE(3)$ is a combination of $\mathbb{R}^3$ and $SO(3)$, where the distortion bound on $\mathbb{R}^3$, as an identity map, is 1. There are different ways to combine the metrics to form a metric on $SE(3)$, but they are all equivalent. Basically, such metrics are defined as, for some $p \geq 1$,

$$d^p_{\mathcal{X}}((\mathbf{x}, \gamma), (\mathbf{y}, \zeta)) = \sqrt[p]{d_{\mathbb{R}^3}(\mathbf{x}, \mathbf{y})^p + d_{SO(3)}(\gamma, \zeta)^p}, \ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^3, \gamma, \zeta \in SO(3),$$

or

$$d^\infty_{\mathcal{X}}((\mathbf{x}, \gamma), (\mathbf{y}, \zeta)) = \max \left\{ d_{\mathbb{R}^3}(\mathbf{x}, \mathbf{y}), d_{SO(3)}(\gamma, \zeta) \right\}, \ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^3, \gamma, \zeta \in SO(3).$$

Their distortions are always controlled by the distortion bounds for the greater component (similar for $d^\infty_{\mathcal{X}}$) since

$$\begin{aligned} d^p_{\mathcal{X}}((\mathbf{x}, \gamma), (\mathbf{y}, \zeta)) &\subseteq \sqrt[p]{d_{\mathbb{R}^3}(\mathbf{x}, \mathbf{y})^p + \mathrm{tort}(\mu^r)^p d_{\widehat{SO}(3)}((\mu^r)^{-1}(\gamma), (\mu^r)^{-1}(\zeta))^p} \\ &\subseteq \sqrt[p]{\mathrm{tort}(\mu^r)^p d_{\mathbb{R}^3}(\mathbf{x}, \mathbf{y})^p + \mathrm{tort}(\mu^r)^p d_{\widehat{SO}(3)}((\mu^r)^{-1}(\gamma), (\mu^r)^{-1}(\zeta))^p} \\ &= \mathrm{tort}(\mu^r) d^p_{\mathcal{W}}((\mathbf{x}, (\mu^r)^{-1}(\gamma)), (\mathbf{y}, (\mu^r)^{-1}(\zeta))) \end{aligned}$$

and to reach the bound, we can choose points in $\mathcal{W}$ such that $\mathbf{x} = \mathbf{y}$ where the distortions are the same with μ^r. Therefore, the distortion of μ is the same with its of μ^r, i.e., $\mathrm{tort}(\mu) = \mathrm{tort}(\mu^r) = [\frac{1}{2}, 2]$. We can take $C_0 = 2$ as our atlas constant in axiom **(A3)** for the representation μ, and we have the Proposition 3.3.1.

Proposition 3.3.1. *The atlas constant of an $SE(3)$ robot is $C_0 = 2$.*

Chapter 4

Delta Footprints and Approximations

Recall that a Delta robot is an isosceles right triangle robot $R_0 = \triangle\mathcal{AOB}$ with its canonical placement $\mathcal{A} = (1,0,0)$, $\mathcal{B} = (0,1,0)$ and $\mathcal{O} = (0,0,0)$. The footprint map for the robot is

$$\begin{aligned} \mathrm{Fp} : \mathbb{R}^3 \times SO(3) &\to \mathcal{C}(\mathbb{R}^3) \\ (\mathbf{x}, \gamma) &\mapsto \{\mathbf{x} + s\gamma(\mathbf{e}_1) + t\gamma(\mathbf{e}_2) : s, t \in [0,1], s + t \leq 1\}, \end{aligned} \tag{4.1}$$

where $\mathbf{e}_1 = (1,0,0) \in \mathbb{R}^3$, $\mathbf{e}_2 = (0,1,0) \in \mathbb{R}^3$. In chapter 2, we have proved that this footprint map Fp is translational. It is continuous due to the continuity in both $\mathbf{x}$ and γ. But this is not enough. The axiom **(A2)** in SSS framework requires the footprint map to be Lipschitz continuous. The axiom **(A0)** requires that the classification of boxes in $\square\mathcal{W}$ are based on an approximate footprint that is σ-effective. In this chapter, we will prove the Lipschitz continuity of Fp and design an approximate footprint that is σ-effective.

4.1 Delta Exact Footprint

The Lipschitz constant of Fp in Equation 4.1 depends on the choice of a metric on $SE(3)$. In the last chapter, we have analyzed the essential metric for $SO(3)$ (i.e., Φ_6 in [26]), and determined the atlas constant $C_0 = 2$ for $SE(3)$ metrics based on Φ_6. In this chapter, we fix the metric for $SE(3)$, say

$$d_{\mathcal{X}}((\mathbf{x},\gamma),(\mathbf{y},\zeta)) = d_{\mathbb{R}^3}(\mathbf{x},\mathbf{y}) + d_{SO(3)}(\gamma,\zeta).$$

Our goal is to estimate the Lipschitz constant and find a heuristic to construct the approximate footprint from the exact footprint of a box in $\square\mathcal{W}$.

4.1.1 Lipschitz Constant for Delta Fp

To estimate the Lipschitz constant, a key problem is to estimate $d_{\mathcal{H}}$ between different $\mathrm{Fp}(\gamma)$ and $\mathrm{Fp}(\zeta)$. Recall that each rotation in $SO(3)$ is determined by an axis $\mathbf{v} \in S^2$ and an angle $\theta \in S^1$, i.e., by Euler-Rodrigues formula [37], the rotation matrix is

$$R(\mathbf{v},\theta) = e^{\theta[\mathbf{v}]} = I + \sin\theta[\mathbf{v}] + (1-\cos\theta)[\mathbf{v}]^2.$$

The metric $d_{SO(3)}$ is defined as $d_{SO(3)}(\gamma,\zeta) = \theta$ for $R(\mathbf{v},\theta) = \gamma\zeta^{-1}$. The rotation by Euler-Rodrigues formula itself restricts the change of the entire space $\mathbb{R}^3$ by Lemma 4.1.1:

Lemma 4.1.1. *For any $R(\mathbf{v},\theta) \in SO(3)$ and any $\mathbf{u} \in \mathbb{R}^3$,*

$$|R(\mathbf{v},\theta)\mathbf{u} - \mathbf{u}| \leq \theta|\mathbf{u}|.$$

Proof. It is sufficient to show that

$$\begin{aligned}
(R(\mathbf{v},\theta) - I)^T(R(\mathbf{v},\theta) - I) &= R(\mathbf{v},\theta)^T R(\mathbf{v},\theta) - (R(\mathbf{v},\theta)^T + R(\mathbf{v},\theta)) + I \\
&= 2I - (R(\mathbf{v},-\theta) + R(\mathbf{v},\theta)) \\
&= 2I - ((I - \sin\theta[\mathbf{v}] + (1-\cos\theta)[\mathbf{v}]^2) + (I + \sin\theta[\mathbf{v}] + (1-\cos\theta)[\mathbf{v}]^2)) \\
&= 2(\cos\theta - 1)[\mathbf{v}]^2
\end{aligned}$$

Note that $[\mathbf{v}]^2 = -\mathrm{diag}\{y^2+z^2, x^2+z^2, x^2+y^2\}$ for $\mathbf{v} = (x,y,z)^T \in S^2$, which is a diagonal matrix with all diagonal entries ranged by $[-1,0]$. Therefore, each eigenvalue of $(R(\mathbf{v},\theta) - I)^T(R(\mathbf{v},\theta) - I)$ is ranged by

$$[0, 2(1-\cos\theta)] \subseteq [0,\theta^2].$$

Hence

$$|R(\mathbf{v},\theta)\mathbf{u}-\mathbf{u}|^2 = \mathbf{u}^T(R(\mathbf{v},\theta)-I)^T(R(\mathbf{v},\theta)-I)\mathbf{u} \le \theta^2\mathbf{u}^T\mathbf{u} = (\theta|\mathbf{u}|)^2,$$

which implies $|R(\mathbf{v},\theta)\mathbf{u}-\mathbf{u}| \le \theta|\mathbf{u}|$. □

Corollary 4.1.2. *For any $\gamma, \zeta \in SO(3)$ and $\mathbf{u} \in \mathbb{R}^3$,*

$$|\gamma(\mathbf{u}) - \zeta(\mathbf{u})| \le d_{SO(3)}(\gamma,\zeta)|\mathbf{u}|.$$

Proof. For $R(\mathbf{v},\theta) = \gamma\zeta^{-1}$, we have

$$|\gamma(\mathbf{u}) - \zeta(\mathbf{u})| = |R(\mathbf{v},\theta)(\zeta(\mathbf{u})) - \zeta(\mathbf{u})| \le \theta|\zeta(\mathbf{u})| = d_{SO(3)}(\gamma,\zeta)|\mathbf{u}|$$

□

Under $d_{\mathcal{X}}$, we can compute the Lipschitz constant L_0 for the Delta footprint, see Theorem 4.1.3:

Theorem 4.1.3. *For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$, and $\gamma, \zeta \in SO(3)$,*

$$d_{\mathcal{H}}(\mathrm{Fp}(\mathbf{x},\gamma), \mathrm{Fp}(\mathbf{y},\zeta)) \le d_{\mathcal{X}}((\mathbf{x},\gamma),(\mathbf{y},\zeta)).$$

The Lipschitz constant L_0 for the Delta Robot is 1.

Proof. Let's begin with the case when $\mathbf{x} = \mathbf{y}$, where

$$d_{\mathcal{X}}((\mathbf{x},\gamma),(\mathbf{x},\zeta)) = d_{SO(3)}(\gamma,\zeta).$$

For each point $\mathbf{p} \in \mathrm{Fp}(\mathbf{x},\gamma)$, $\mathbf{p} = \mathbf{x} + s\gamma(\mathbf{e}_1) + t\gamma(\mathbf{e}_2)$ for some $s, t \in [0,1]$, $s+t \le 1$. Note that γ is linear, so

$$\mathbf{p} = \mathbf{x} + \gamma(s\mathbf{e}_1 + t\mathbf{e}_2),$$

and there is $\mathbf{q} = \mathbf{x} + \zeta(s\mathbf{e}_1 + t\mathbf{e}_2) \in \mathrm{Fp}(\mathbf{x},\zeta)$, such that

$$|\mathbf{p}-\mathbf{q}| = |\gamma(s\mathbf{e}_1+t\mathbf{e}_2) - \zeta(s\mathbf{e}_1+t\mathbf{e}_2)| \le d_{SO(3)}(\gamma,\zeta)|s\mathbf{e}_1+t\mathbf{e}_2| \le d_{SO(3)}(\gamma,\zeta).$$

Thus $\forall \mathbf{p} \in \mathrm{Fp}(\mathbf{x},\gamma)$,

$$\mathrm{Sep}(\{\mathbf{p}\}, \mathrm{Fp}(\mathbf{x},\zeta)) \le |\mathbf{p}-\mathbf{q}| \le d_{SO(3)}(\gamma,\zeta),$$

which implies

$$\sup_{\mathbf{p}\in \mathrm{Fp}(\mathbf{x},\gamma)} \mathrm{Sep}(\{\mathbf{p}\}, \mathrm{Fp}(\mathbf{x},\zeta)) \le d_{SO(3)}(\gamma,\zeta).$$

Similar,

$$\sup_{\mathbf{q}\in \mathrm{Fp}(\mathbf{x},\zeta)} \mathrm{Sep}(\{\mathbf{q}\}, \mathrm{Fp}(\mathbf{x},\gamma)) \le d_{SO(3)}(\gamma,\zeta).$$

Therefore

$$d_{\mathcal{H}}(\mathrm{Fp}(\mathbf{x},\gamma), \mathrm{Fp}(\mathbf{x},\zeta)) \le d_{SO(3)}(\gamma,\zeta).$$

In the end, for $\mathbf{x} \neq \mathbf{y}$, we have the triangle inequality:

$$\begin{aligned} d_{\mathcal{H}}(\mathrm{Fp}(\mathbf{x},\gamma), \mathrm{Fp}(\mathbf{y},\zeta)) &\le d_{\mathcal{H}}(\mathrm{Fp}(\mathbf{x},\gamma), \mathrm{Fp}(\mathbf{x},\zeta)) + d_{\mathcal{H}}(\mathrm{Fp}(\mathbf{x},\zeta), \mathrm{Fp}(\mathbf{y},\zeta)) \\ &\le d_{SO(3)}(\gamma,\zeta) + d_{\mathbb{R}^3}(\mathbf{x},\mathbf{y}) \\ &= d_{\mathcal{X}}((\mathbf{x},\gamma),(\mathbf{y},\zeta)). \end{aligned}$$

□

Remark 4.1.4. An enlarged Delta robot R_0^λ is a footprint map

$$\begin{aligned} &\mathrm{Fp} : SE(3) \to \mathcal{C}(\mathbb{R}^3) \\ &\quad (\mathbf{x},\gamma) \mapsto \{\mathbf{x} + \lambda(s\gamma(\mathbf{e}_1) + t\gamma(\mathbf{e}_2)) : s,t \in [0,1], s+t \le 1\}, \end{aligned}$$

for some $\lambda > 0$. Its canonical placement is an enlarged triangle with lengths of the legs becoming λ. It is easy to show that the Lipschitz constant for R_0^λ is $\max\{1,\lambda\}$ by the Corollary 4.1.2.

4.1.2 Exact Footprint for a Box

It is not easy to characterize the exact footprint for a box $B = B^t \times B^r \in \square\mathcal{W}$. But we can get some approximations from its behaviour along line segments. Let us first get rid of the translational component B^t which only affects the footprint by a Minkowski sum. We only consider the footprints for $\mathrm{Fp}(B^r)$, which by definition, is $\mathrm{Fp}(\{\mathbf{0}\} \times B^r)$. Note that given a box B^r and any points $\mathbf{p},\mathbf{q} \in B^r$, the line segment $\overline{\mathbf{pq}}$ is the shortest curve connecting $\mathbf{p}$ and $\mathbf{q}$, which is the one that defines $d_{\widehat{SO}(3)}(\mathbf{p},\mathbf{q})$. Then

$\mu(\overline{\mathbf{pq}})$ is correspondingly the shortest curve connecting $\gamma = \mu(\mathbf{p})$ and $\zeta = \mu(\mathbf{q})$, which is the curve given by the single exponential map connecting γ and ζ, i.e.,

$$\mu(\overline{\mathbf{pq}}) = e^{t \log(\gamma\zeta^{-1})}\zeta, \ t \in [0,1].$$

The exponential map defines the footprint by rotating the robot along a fixed axis. We can classify the footprints $\mathrm{Fp}(\overline{\mathbf{pq}})$ by the different cases of a triangle rotating along different fixed axes; see Example C.0.1 in the appendix. For any $\mathbf{x} \in \mathbb{R}^3$, let $\mathrm{Fp}_{\mathbf{x}}$ be the footprint map defined by

$$\begin{aligned} \mathrm{Fp}_{\mathbf{x}} : SO(3) &\to \mathcal{C}(\mathbb{R}^3) \\ \gamma &\mapsto \{\gamma(\mathbf{x})\}, \end{aligned}$$

i.e., the footprint of purely point $\mathbf{x}$. As a lemma to be used later, we summarize the behaviour of footprints of a single point, see Lemma 4.1.5.

Lemma 4.1.5. *Let $B^r \in \square\mathcal{W}^r$. For any $\mathbf{p}, \mathbf{q} \in B^r$, if a ball $Ball(\mathbf{o}, s) \subseteq \mathbb{R}^3$ contains both $\mathrm{Fp}_{\mathbf{x}}(\mathbf{p})$ and $\mathrm{Fp}_{\mathbf{x}}(\mathbf{q})$, such that the radius $s < |\mathbf{x}|$, then*

$$\mathrm{Fp}_{\mathbf{x}}(\overline{\mathbf{pq}}) \subseteq Ball(\mathbf{o}, s).$$

Proof. By the deductions on the effect of μ^r on shortest curves, $\mu^r(\overline{\mathbf{pq}})$ is the arc of a circle center at the origin of $\mathbb{R}^3$ with radius $|\mathbf{x}|$, connecting $\mu^r(\mathbf{p})$ and $\mu^r(\mathbf{q})$. This arc is always contained in such balls $\mathrm{Ball}(\mathbf{o}, s)$ when $s < |\mathbf{x}|$. $\square$

When applying Lemma 4.1.5 to point $\mathcal{A}$, i.e., $\mathbf{x} = \mathbf{e}_1$, and point $\mathcal{B}$, i.e., $\mathbf{x} = \mathbf{e}_2$, balls with radius less than 1 that includes the footprints of endpoints can always include the whole arc of footprints of $\mathcal{A}$ or $\mathcal{B}$. A box B^r in $\widehat{SO}(3)$ is the convex hull of its 8 corners $\mathbf{b}_i$ for $i = 1, \ldots, 8$. If we collect the footprints $\mathrm{Fp}_{\mathcal{A}}$ and $\mathrm{Fp}_{\mathcal{B}}$ (identifying by representation μ^r) at the 8 corners and use balls $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ to contain them respectively, then the footprints of any connection $\mathrm{Fp}_{\mathcal{A}}(\overline{b_i b_j})$ and $\mathrm{Fp}_{\mathcal{B}}(\overline{b_i b_j})$ will also be contained in the corresponding balls, as well as the footprints of connections between points on the connections, i.e., the whole box. Then since the robot $R_0 = \triangle\mathcal{AOB}$ is also convex, the convex hull of $\mathcal{O}$ and those balls will include the footprints $\mathrm{Fp}_{\mathbf{x}}(\mathbf{b})$ of all points $\mathbf{x}$ in the robot R_0 for any $\mathbf{b} \in B^r$. This is a heuristic for our approximate footprint. We develop its algebraic construction in the next section.

4.2 Delta Approximate Footprint

In this section, we develop the algebraic construction of the approximate footprint for the Delta robot R_0 on a given box $B \in \square\mathcal{W}$. As mentioned, Fp is translational and so we can begin with $\mathrm{Fp}(B^r)$ for $B^r \in \square\mathcal{W}^r$.

4.2.1 $SO(3)$ approximate footprint

For $B^r = \widehat{SO}(3)$, the footprint $\mathrm{Fp}(\widehat{SO}(3))$ is the whole ball Ball(1), and we define it as $\widetilde{\mathrm{Fp}}(\widehat{SO}(3))$. We consider when $B^r \neq \widehat{SO}(3)$.

Our $\widetilde{\mathrm{Fp}}(B^r)$ is constructed by the following process. Note that $\mathcal{A}$ is $\mathbf{e}_1$ and $\mathcal{B}$ is $\mathbf{e}_2$:

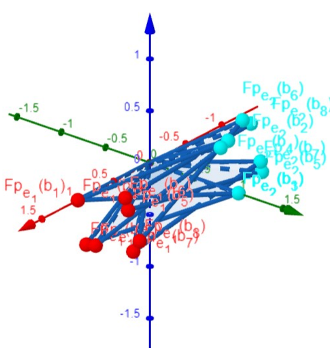


Figure 4.1: Compute clusters.

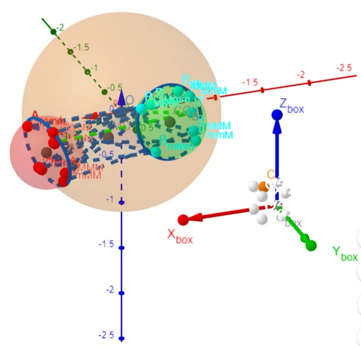


Figure 4.2: Construct ball approximations.

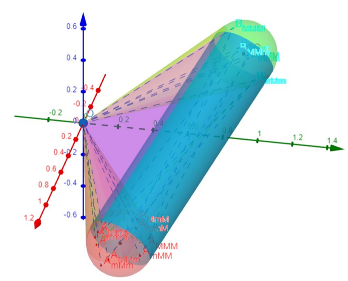


Figure 4.3: Make convex hull.

Design of $\widetilde{\mathrm{Fp}}(B^r)$

- Compute the 8 corners of B^r, denote them by $\mathbf{b}_i$ for $i = 1, \ldots, 8$, construct $\mathrm{Fp}_{\mathbf{e}_k}(\mathbf{b}_i)$ for $i = 1, \ldots, 8$ and $j = 1, 2$. Each of the sets $\{\mathrm{Fp}_{\mathbf{e}_k}(\mathbf{b}_i) : i = 1, \ldots, 8\}$ is called a **cluster**. It is $\mathcal{A}$-cluster if $j = 1$ and is $\mathcal{B}$-cluster if $j = 2$. See Figure 4.1.

- Let $\mathrm{m}(\mathbf{e}_k, B^r) = \frac{1}{8}\sum_i \mathrm{Fp}_{\mathbf{e}_k}(\mathbf{b}_i)$ for $j = 1, 2$, and

$$\mathrm{d}(B^r) = \max_{i=1,\ldots,8,j=1,2} \left\{\mathrm{Fp}_{\mathbf{e}_k}(\mathbf{b}_i) - \mathrm{m}(\mathbf{e}_k, B^r)\right\},$$

define two balls $\widetilde{\mathrm{Fp}}_{\mathbf{e}_k}(B^r) = \mathrm{Ball}(\mathrm{m}(\mathbf{e}_k, B^r), \mathrm{d}(B^r))$ containing each cluster respectively. See Figure 4.2.

- Define the approximate footprint to be the convex hull of two balls and $\mathbf{0}$, i.e.,

$$\widetilde{\mathrm{Fp}}(B^r) = \mathrm{Chull}\{\widetilde{\mathrm{Fp}}_{\mathbf{e}_1}(B^r), \widetilde{\mathrm{Fp}}_{\mathbf{e}_2}(B^r), \mathbf{0}\}.$$

See Figure 4.3.

The behavior of $\widetilde{\mathrm{Fp}}(B^r)$ in the process designed above is the following:

Recall that the child of $\widehat{SO}(3)$ are $C_w = \{(-1, x, y, z) : x, y, z \in [-1, 1]\}$, $C_x = \{(w, -1, y, z) : w, y, z \in [-1, 1]\}$, $C_y = \{(w, x, -1, z) : w, x, z \in [-1, 1]\}$ and $C_z = \{(w, x, y, -1) : w, x, y \in [-1, 1]\}$. The $\widetilde{\mathrm{Fp}}$ for these four boxes are the same. They are still Ball(1), since they share the same 8 corners. When B^r is half size of the C_w, C_x, C_y and C_z, the $\widetilde{\mathrm{Fp}}(B^r)$ is only the convex hull of the two balls $\widetilde{\mathrm{Fp}}_{\mathbf{e}_k}(B^r)$, see Figure 4.4.. When B^r is smaller, the $\widetilde{\mathrm{Fp}}$ is the shape in Figure 4.3.

4.2.2 $SE(3)$ approximate footprint

Since Fp is translational, $\mathrm{Fp}(B^t \times B^r) = B^t \oplus \mathrm{Fp}(B^r)$. To approximate B^t, we simply use the circumscribed ball of B^t, denoted by $\mathrm{Ball}(B^t) = \mathrm{Ball}(\mathrm{m}(B^t), \mathrm{r}(B^t))$, where $\mathrm{r}(B^t) = \frac{1}{2}\ell(B^t)$. Then the

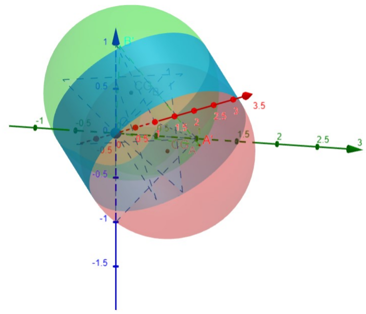


Figure 4.4: Degenerate case.

approximate footprint for $B = B^t \times B^r$ is

$$\widetilde{\mathrm{Fp}}(B) = \mathrm{Ball}(B^t) \oplus \widetilde{\mathrm{Fp}}(B^r).$$

We have an explicit expression for $\widetilde{\mathrm{Fp}}(B)$ by Theorem 4.2.1 based on the lemma of Minkowski sum of convex hulls, see Lemma A.4.3.

Theorem 4.2.1. *For $B = B^t \times B^r$,*

$$\widetilde{\mathrm{Fp}}(B) = \mathit{Chull}\left\{\mathit{Ball}\left(\mathit{m}(\mathbf{e}_1, B^r) + \mathit{m}(B^t), \mathit{d}(B^r) + \mathit{r}(B^t)\right), \mathit{Ball}\left(\mathit{m}(\mathbf{e}_2, B^r) + \mathit{m}(B^t), \mathit{d}(B^r) + \mathit{r}(B^t)\right), \mathit{Ball}(B^t)\right\}.$$

Proof.

$$\begin{aligned}
\widetilde{\mathrm{Fp}}(B) &= \mathrm{Ball}(B^t) \oplus \widetilde{\mathrm{Fp}}(B^r) \\
&= \mathrm{Ball}(\mathrm{m}(B^t), \mathrm{r}(B^t)) \oplus \mathrm{Chull}\left\{\widetilde{\mathrm{Fp}}_{\mathbf{e}_1}(B^r), \widetilde{\mathrm{Fp}}_{\mathbf{e}_2}(B^r), \mathbf{0}\right\} \\
&= \mathrm{Ball}(\mathrm{m}(B^t), \mathrm{r}(B^t)) \oplus \mathrm{Chull}\left\{\mathrm{Ball}(\mathrm{m}(\mathbf{e}_1, B^r), \mathrm{d}(B^r)), \mathrm{Ball}(\mathrm{m}(\mathbf{e}_2, B^r), \mathrm{d}(B^r)), \mathbf{0}\right\} \\
&= \mathrm{Chull}\left\{\mathrm{Ball}(\mathrm{m}(B^t), \mathrm{r}(B^t)) \oplus \mathrm{Ball}(\mathrm{m}(\mathbf{e}_1, B^r), \mathrm{d}(B^r)),\right. \\
&\qquad \mathrm{Ball}(\mathrm{m}(B^t), \mathrm{r}(B^t)) \oplus \mathrm{Ball}(\mathrm{m}(\mathbf{e}_2, B^r), \mathrm{d}(B^r)), \\
&\qquad \left.\mathrm{Ball}(\mathrm{m}(B^t), \mathrm{r}(B^t))\right\} \\
&= \mathrm{Chull}\left\{\mathrm{Ball}\left(\mathrm{m}(\mathbf{e}_1, B^r) + \mathrm{m}(B^t), \mathrm{d}(B^r) + \mathrm{r}(B^t)\right), \mathrm{Ball}\left(\mathrm{m}(\mathbf{e}_2, B^r) + \mathrm{m}(B^t), \mathrm{d}(B^r) + \mathrm{r}(B^t)\right), \mathrm{Ball}(B^t)\right\}
\end{aligned}$$

$\square$

4.2.3 σ-effectiveness of $\widetilde{\mathrm{Fp}}$ for Delta footprint

In this section we prove the σ-effectiveness of the $\widetilde{\mathrm{Fp}}$ defined above. We summarize the result as Theorem 4.2.2.

Theorem 4.2.2. *$\widetilde{\mathrm{Fp}}(B) = Ball(B^t) \oplus \widetilde{\mathrm{Fp}}(B^r)$ is σ-effective with $\sigma = \frac{3}{2}D_0$, where D_0 is the bound in axiom **(A1)** for aspect ratios of boxes in $\square\mathcal{W}$.*

Proof. For the conservativeness of $\widetilde{\mathrm{Fp}}(B)$, we first consider the conservativeness of $\widetilde{\mathrm{Fp}}(B^r)$. Note that, for any pair of corners $\mathbf{b}_i, \mathbf{b}_j$ of B^r, $\mathrm{Fp}_{\mathbf{e}_k}(\mathbf{b}_i) \in \widetilde{\mathrm{Fp}}_{\mathbf{e}_k}(B^r)$ and $\mathrm{Fp}_{\mathbf{e}_k}(\mathbf{b}_j) \in \widetilde{\mathrm{Fp}}_{\mathbf{e}_k}(B^r)$ for $k = 1, 2$. By Lemma 4.1.5, $\forall \mathbf{p} \in \overline{\mathbf{b}_i\mathbf{b}_j}$, $\mathrm{Fp}_{\mathbf{e}_k}(\mathbf{p}) \in \widetilde{\mathrm{Fp}}_{\mathbf{e}_k}(B^r)$. Then for any $\mathbf{b} \in B^r$, there is $\mathbf{p} \in \overline{\mathbf{b}_i\mathbf{b}_j}$ and $\mathbf{q} \in \overline{\mathbf{b}_l\mathbf{b}_m}$ such that $\mathbf{b} \in \overline{\mathbf{pq}}$, which also by Lemma 4.1.5, implies that $\mathrm{Fp}_{\mathbf{e}_k}(\mathbf{b}) \in \widetilde{\mathrm{Fp}}_{\mathbf{e}_k}(B^r)$ for $k = 1, 2$, and hence $\mathrm{Fp}_{\mathbf{e}_k}(B^r) \subseteq \widetilde{\mathrm{Fp}}_{\mathbf{e}_k}(B^r)$ for $k = 1, 2$. By linearity of $SO(3)$, for any $\mathbf{x} = s\mathbf{e}_1 + t\mathbf{e}_2$ such that $s, t \in [0, 1]$, $s + t \leq 1$, and any $\mathbf{b} \in B^r$,

$$\mathrm{Fp}_{\mathbf{x}}(\mathbf{b}) = s\mathrm{Fp}_{\mathbf{e}_1}(\mathbf{b}) + t\mathrm{Fp}_{\mathbf{e}_2}(\mathbf{b}) \in \mathrm{Chull}\left\{\mathbf{0}, \mathrm{Fp}_{\mathbf{e}_1}(\mathbf{b}), \mathrm{Fp}_{\mathbf{e}_2}(\mathbf{b})\right\}.$$

Therefore

$$\begin{aligned}
\mathrm{Fp}(B^r) &= \cup_{\mathbf{x}=s\mathbf{e}_1+t\mathbf{e}_2}\mathrm{Fp}_{\mathbf{x}}(B^r) \\
&\subseteq \mathrm{Chull}\left\{\mathbf{0}, \mathrm{Fp}_{\mathbf{e}_1}(B^r), \mathrm{Fp}_{\mathbf{e}_2}(B^r)\right\} \\
&\subseteq \mathrm{Chull}\left\{\mathbf{0}, \widetilde{\mathrm{Fp}}_{\mathbf{e}_1}(B^r), \widetilde{\mathrm{Fp}}_{\mathbf{e}_2}(B^r)\right\} \\
&= \widetilde{\mathrm{Fp}}(B^r).
\end{aligned}$$

Then for any $B = B^t \times B^r$,

$$\mathrm{Fp}(B) = B^t \oplus \mathrm{Fp}(B^r) \subseteq \mathrm{Ball}(B^t) \oplus \widetilde{\mathrm{Fp}}(B^r) = \widetilde{\mathrm{Fp}}(B),$$

which proves that $\widetilde{\mathrm{Fp}}$ is conservative.

For the effectiveness of $\widetilde{\mathrm{Fp}}(B)$, be aware that we have proved that Fp and μ satisfy axioms **(A2)** and **(A3)** with $L_0 = 1$ and $C_0 = 2$, so as $\mathrm{Fp}_{\mathbf{e}_k}$ for $k = 1, 2$. We have for $B = B^t \times B^r$,

$$
\begin{aligned}
\mathrm{d}(B^r) &\le \frac{1}{2} \sup_{\mathbf{p},\mathbf{q}\in B^r, k=1,2} d_{\mathcal{H}}(\mathrm{Fp}_{\mathbf{e}_k}(\mathbf{p}), \mathrm{Fp}_{\mathbf{e}_k}(\mathbf{q})) \\
&\le \frac{1}{2} L_0 \sup_{\mathbf{p}\mathbf{q}\in B^r} d_{SO(3)}(\mu^r(\mathbf{p}), \mu^r(\mathbf{q})) \\
&\le \frac{1}{2} L_0 C_0 \sup_{\mathbf{p}\mathbf{q}\in B^r} d_{\widehat{SO}(3)}(\mathbf{p}, \mathbf{q}) \\
&\le \frac{1}{2} L_0 C_0 \ell(B^r) \\
&\le \frac{1}{2} L_0 C_0 \ell(B) \\
&\le \frac{1}{2} L_0 C_0 D_0 \mathrm{w}(B) \\
&\le \frac{1}{2} L_0 C_0 D_0 \mathrm{w}(B^t),
\end{aligned}
$$

which implies $\widetilde{\mathrm{Fp}}_{\mathbf{e}_k}(B^r) \subseteq \mathrm{Ball}(\mathrm{m}(\mathbf{e}_k, B^r), \frac{1}{2} L_0 C_0 D_0 \mathrm{w}(B^t))$ for $k = 1, 2$. Hence

$$
\begin{aligned}
\widetilde{\mathrm{Fp}}(B^r) &= \mathrm{Chull}\left\{\widetilde{\mathrm{Fp}}_{\mathbf{e}_1}(B^r), \widetilde{\mathrm{Fp}}_{\mathbf{e}_2}(B^r), \mathbf{0}\right\} \\
&\subseteq \mathrm{Chull}\left\{\mathrm{Ball}\left(\mathrm{m}(\mathbf{e}_1, B^r), \frac{1}{2} L_0 C_0 D_0 \mathrm{w}(B^t)\right), \mathrm{Ball}\left(\mathrm{m}(\mathbf{e}_2, B^r), \frac{1}{2} L_0 C_0 D_0 \mathrm{w}(B^t)\right), \mathbf{0}\right\} \\
&\subseteq \mathrm{Chull}\left\{\mathrm{m}(\mathbf{e}_1, B^r), \mathrm{m}(\mathbf{e}_2, B^r), \mathbf{0}\right\} \oplus \mathrm{Ball}\left(\frac{1}{2} L_0 C_0 D_0 \mathrm{w}(B^t)\right) \\
&\subseteq \mathrm{Fp}(B^r) \oplus \mathrm{Ball}\left(\frac{1}{2} L_0 C_0 D_0 \mathrm{w}(B^t)\right).
\end{aligned}
$$

Therefore

$$
\begin{aligned}
\widetilde{\mathrm{Fp}}(B) &= \mathrm{Ball}(B^t) \oplus \widetilde{\mathrm{Fp}}(B^r) \\
&= \mathrm{Ball}\left(\mathrm{m}(B^t), \frac{1}{2}\ell(B^t)\right) \oplus \widetilde{\mathrm{Fp}}(B^r) \\
&\subseteq \mathrm{Ball}\left(\mathrm{m}(B^t), \frac{1}{2} D_0 \mathrm{w}(B^t)\right) \oplus \mathrm{Ball}\left(\frac{1}{2} L_0 C_0 D_0 \mathrm{w}(B^t)\right) \oplus \mathrm{Fp}(B^r) \\
&= \mathrm{Ball}\left(\mathrm{m}(B^t), \frac{1}{2}(L_0 C_0 + 1) D_0 \mathrm{w}(B^t)\right) \oplus \mathrm{Fp}(B^r) \\
&\subseteq \left(\frac{1}{2}(L_0 C_0 + 1) D_0 B^t\right) \oplus \mathrm{Fp}(B^r) \\
&\subseteq \left(\frac{1}{2}(L_0 C_0 + 1) D_0 B^t\right) \oplus \mathrm{Fp}\left(\frac{1}{2}(L_0 C_0 + 1) D_0 B^r\right) \\
&= \mathrm{Fp}\left(\frac{1}{2}(L_0 C_0 + 1) D_0 B\right)
\end{aligned}
$$

We plug in $L_0 = 1$ and $C_0 = 2$, which results in

$$\widetilde{\mathrm{Fp}}(B) \subseteq \mathrm{Fp}\left(\frac{3}{2} D_0 B\right),$$

which proves that $\widetilde{\mathrm{Fp}}$ is $\frac{3}{2}D_0$-effective. $\square$

Chapter 5

Explicit Parameterized Collision Detection

Recall that the classification of soft predicates is given by the method of features [56], defined by

$$\widetilde{C}(B) = \begin{cases} \texttt{FREE} & \text{if } \widetilde{\phi}(B) = \emptyset,\ \exists \mathbf{b} \in \widetilde{\mathrm{Fp}}(B), \mathbf{b} \notin \Omega \\ \texttt{STUCK} & \text{if } \widetilde{\phi}(B) = \emptyset,\ \exists \mathbf{b} \in \widetilde{\mathrm{Fp}}(B), \mathbf{b} \in \Omega \\ \texttt{MIXED} & \text{if } \widetilde{\phi}(B) \neq \emptyset \end{cases},$$

where

$$\widetilde{\phi}(B) = \{f \in \Phi(\Omega) : f \cap \widetilde{\mathrm{Fp}}(B) \neq \emptyset\}.$$

The process of maintaining $\widetilde{\phi}(B)$ has been described in [56], which is the following:

Maintaining $\widetilde{\phi}(B)$

- Initialize $\widetilde{\phi}(B_0)$ by $\Phi(B)$.
- After each split, for each child B from parent(B), initialize $\widetilde{\phi}(B)$ by $\emptyset$. For each $f \in \widetilde{\phi}(\text{parent}(B))$, check if $f \cap \widetilde{\mathrm{Fp}}(B) \neq \emptyset$. If so, insert f into $\widetilde{\phi}(B)$. Repeat this step for all $f \in \widetilde{\phi}(\text{parent}(B))$.

The correctness of the process is based on two aspects. The first is $\widetilde{\phi}(B) \subseteq \widetilde{\phi}(\text{parent}(B))$ for any $B \in \square\mathcal{W}$, which we call the **inheritance of feature set**. The second is the criterion for $f \cap \widetilde{\mathrm{Fp}}(B) \neq \emptyset$, or equivalently, $\mathrm{Sep}(f, \widetilde{\mathrm{Fp}}(B)) > 0$. The first technique is based on the subdivision scheme, which we will develop in the next chapter. The second technique is solving the inequality $\mathrm{Sep}(f, \widetilde{\mathrm{Fp}}(B)) > 0$ exactly. The solution for solving this inequality requires a semi-algebraic representation for $\widetilde{\mathrm{Fp}}(B)$ which turns the inequality into solving polynomial inequalities. Note that the latter requires a solution for the "Zero Problem" which by [61] is not practical to implement for polynomials with high degrees. It is important to reduce the degree of all numbers that are operated throughout the SSS planning, at most degree 4, over $\mathbb{Q}$.

In this chapter, we will begin with the Σ_2 decomposition for $\widetilde{\mathrm{Fp}}(B)$ to give it a semi-algebraic representation, and develop an explicit process for collision detections, i.e., checking if $\mathrm{Sep}(f, \widetilde{\mathrm{Fp}}(B)) > 0$.

5.1 On the Σ_2 Decomposition Technique

An **algebraic set** $\overline{\mathcal{I}}$ is a set of points that satisfy a set of algebraic equations. A **semi-algebraic set** $\mathcal{I} \subseteq \mathbb{R}^k$ is an extension to the algebraic sets that allows the constraints to include inequalities [30]. $\mathcal{I}$ is **simple** if there is an algebraic set $\overline{\mathcal{I}}$ such that $\mathcal{I} \subseteq \overline{\mathcal{I}}$ and $\dim(\mathcal{I}) = \dim(\overline{\mathcal{I}})$. We call $\overline{\mathcal{I}}$ the **algebraic span** of $\mathcal{I}$. Our feature set $\Phi(\Omega)$ consist of simple sets such as a point, which is called a **corner** feature, a line segment, which is called an **edge** feature, a triangle, which is called a **facet** feature, where their algebraic spans are point/line/plane respectively.

The technique of Σ_2 decomposition is based on expressing $\widetilde{\mathrm{Fp}}$ as a Σ_2 set, which was first introduced in [23]. A Σ_2 set is a finite union of intersections of elementary sets. We say a set $\mathcal{K} \subseteq \mathbb{R}^3$ is **elementary** if $\mathcal{K} = \{\mathbf{x} \in \mathbb{R}^3 : f(\mathbf{x}) \leq 0\}$ for some polynomial $f : \mathbb{R}^3 \to \mathbb{R}$ of total degree at most 2, and the coefficients of f are algebraic numbers. Examples of elementary sets include half-spaces, infinite cylinders, doubly-infinite cones, ellipsoids, hyperbolas, etc. A **Π_1-set** is a finite intersection of elementary sets and a **Σ_2-set** is a finite union of Π_1 sets.

5.1.1 Σ_2 decomposition for $\widetilde{\mathrm{Fp}}(B)$

Our $\widetilde{\mathrm{Fp}}(B)$ for Delta robot is decomposed into a union of Π_1 sets in Example 5.1.1.

Example 5.1.1. The decomposition is the following:

$\widetilde{\mathrm{Fp}}(B)$ as union of Π_1 sets

- Each $\widetilde{\mathrm{Fp}}_{\mathbf{e}_j}(B) = \mathrm{Ball}(B^t) \oplus \widetilde{\mathrm{Fp}}_{\mathbf{e}_j}(B^r) = \mathrm{Ball}(\mathrm{m}(\mathbf{e}_j, B^r) + \mathrm{m}(B^t), \mathrm{d}(B^r) + \mathrm{r}(B^t))$ for $j = 1, 2$ is a ball that is elementary. $\widetilde{\mathrm{Fp}}_{\mathbf{0}}(B) = \mathrm{Ball}(B^t)$ is also a ball that is elementary.

  We denote $\mathrm{m}_{\mathcal{A}}(B) = \mathrm{m}(\mathbf{e}_1, B^r) + \mathrm{m}(B^t)$, $\mathrm{m}_{\mathcal{B}}(B) = \mathrm{m}(\mathbf{e}_2, B^r) + \mathrm{m}(B^t)$, $\mathrm{d}(B) = \mathrm{d}(B^r) + \mathrm{r}(B^t)$ and $\mathrm{r}(B) = \mathrm{r}(B^t)$.

  And also $\mathrm{S}_{\mathcal{A}}(B) = \widetilde{\mathrm{Fp}}_{\mathbf{e}_1}(B) = \mathrm{Ball}(\mathrm{m}_{\mathcal{A}}(B), \mathrm{d}(B))$, $\mathrm{S}_{\mathcal{B}}(B) = \widetilde{\mathrm{Fp}}_{\mathbf{e}_2}(B) = \mathrm{Ball}(\mathrm{m}_{\mathcal{B}}(B), \mathrm{d}(B))$, $\mathrm{S}_{\mathcal{O}}(B) = \widetilde{\mathrm{Fp}}_{\mathbf{0}}(B) = \mathrm{Ball}(\mathrm{m}(B^t), \mathrm{r}(B))$. See Figure 5.1.

- There is a cylinder $\mathrm{Cylinder}(B)$ whose union with $\mathrm{S}_{\mathcal{A}}(B)$ and $\mathrm{S}_{\mathcal{B}}(B)$ is $\mathrm{Chull}\{\mathrm{S}_{\mathcal{A}}(B), \mathrm{S}_{\mathcal{B}}(B)\}$. See Figure 5.2.

- There are two frustums $\mathrm{Frustum}_{\mathcal{A}}(B)$ and $\mathrm{Frustum}_{\mathcal{B}}(B)$ such that $\mathrm{S}_{\mathcal{O}}(B) \cup \mathrm{Frustum}_{\mathcal{A}}(B) \cup \mathrm{S}_{\mathcal{A}}(B) = \mathrm{Chull}\{\mathrm{S}_{\mathcal{O}}(B), \mathrm{S}_{\mathcal{A}}(B)\}$ and $\mathrm{S}_{\mathcal{O}}(B) \cup \mathrm{Frustum}_{\mathcal{B}}(B) \cup \mathrm{S}_{\mathcal{B}}(B) = \mathrm{Chull}\{\mathrm{S}_{\mathcal{O}}(B), \mathrm{S}_{\mathcal{B}}(B)\}$. See Figure 5.3.

- There is a pyramid $\mathrm{Pyramid}(B)$ whose union with all other Π_1 sets forms the convex hull $\mathrm{Chull}\{\mathrm{S}_{\mathcal{O}}(B), \mathrm{S}_{\mathcal{A}}(B), \mathrm{S}_{\mathcal{B}}(B)\}$. See Figure 5.4.

- The union of all Π_1 sets above is $\widetilde{\mathrm{Fp}}(B)$. See Figure 5.5.

The Σ_2 decomposition for approximate footprint for $B \in \square\mathcal{W}$ of the Delta robot is then

$$\widetilde{\mathrm{Fp}}(B) = \mathrm{S}_{\mathcal{O}}(B) \cup \mathrm{S}_{\mathcal{A}}(B) \cup \mathrm{S}_{\mathcal{B}}(B) \cup \mathrm{Cylinder}(B) \cup \mathrm{Frustum}_{\mathcal{A}}(B) \cup \mathrm{Frustum}_{\mathcal{B}}(B) \cup \mathrm{Pyramid}(B).$$

The algebraic inequalities defining each Π_1 set in Example 5.1.1 are put into Examples D.1.1, D.1.2 and D.1.3 in the appendix.

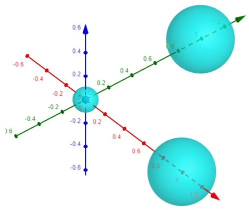


Figure 5.1: The balls.

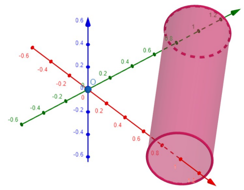


Figure 5.2: The cylinder.

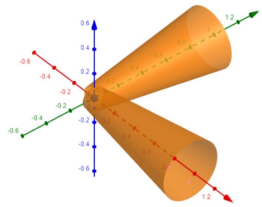


Figure 5.3: The frustums.

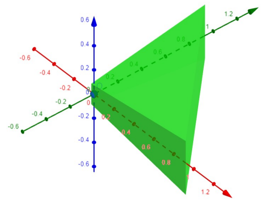


Figure 5.4: The pyramid.

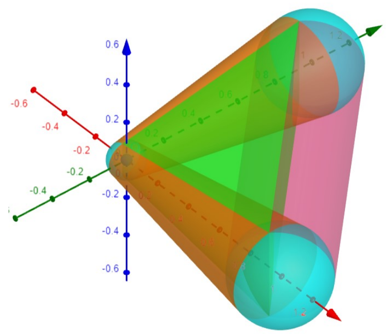


Figure 5.5: The $\widetilde{\mathrm{Fp}}(B)$.

5.1.2 Σ_2 double loop

A Σ_2 set $\mathcal{I}$ can be written as

$$\mathcal{I} = \bigcup_{i=1}^{m} \bigcap_{j=1}^{n} \mathcal{K}_{ij},$$

where $\mathcal{K}_{ij}$ are elementary sets. Each $\mathcal{J}_i = \cap_{j=1}^{n} \mathcal{K}_{ij}$ is an Π_1 set. There is a simple double loop that can answer if $f \cap \mathcal{I} = \emptyset$ for any feature f:

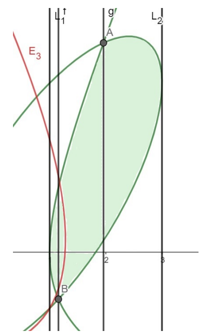


Figure 5.6: CAD method for maintaining intersection $\mathcal{J}$.

Σ_2-Collision Detection(f,A)

Input: f and $\mathcal{I} = \cup_{i=1}^{m} \cap_{j=1}^{n} \mathcal{K}_{ij}$.

Output: `success` if $\mathcal{I} \cap f = \emptyset$, `failure` else.

For $i = 1$ to m

  $\mathcal{J} \leftarrow f$

  For $j = 1$ to n

    $\mathcal{J} \leftarrow \mathcal{J} \cap \mathcal{K}_{ij}$ (*)

    If $\mathcal{J} = \emptyset$, break

  If $\mathcal{J} \neq \emptyset$, return `failure`

Return `success`

The step $(*)$ maintains $\mathcal{J}$ as the intersection of f with successive primitives. If f is a corner or an edge, this is trivial. When f is a facet, this could still be solved in the previous paper for rod and ring robots [23]. But for the Delta robot $R_0 = \triangle\mathcal{AOB}$, maintaining a planar set bounded by curves of degree 2 is required. We can use cylindrical algebraic decomposition (CAD) [30] to exactly maintain the set $\mathcal{J}$, see Figure 5.6 for a reference, but the complexity of algebraic numbers are not easily to be controlled when many planar quadratic equations are involved. We could also use linear approximations of $\mathcal{J}$ to maintain it with an effectiveness, see definition D.2.1 and examples D.2.2, D.2.4 and D.2.6 and their corresponding propositions in Section D.2 for a reference. But it involves additional approximations other than the soft predicates in the SSS framework which plugs in more effective ratios to affect the resolution constant. In fact, we have a much more concise technique that only requires solving explicit quadratic equations without any approximations to exactly decide if $\mathrm{Sep}(f, \widetilde{\mathrm{Fp}}(B)) > 0$. This technique is the so-called parametric separation query and it is endowed with a boundary reduction method.

5.2 Boundary Reduction and Parametric Query

We write $\widetilde{\mathrm{Fp}}(B) = \cup_i \mathcal{J}_i$ where each $\mathcal{J}_i$ is a Π_1 set defined in Example 5.1.1. The statement $\mathrm{Sep}(f, \widetilde{\mathrm{Fp}}(B)) > 0$ is equivalent to $\mathrm{Sep}(f, \mathcal{J}_i) > 0$ for all i. The technique to decide the statement is the parametric separation query solved by boundary reduction method.

A **parametric separation query** is a query asking if $\mathrm{Sep}(\mathcal{I}, \mathcal{K}) > s$ for some $\mathcal{I}, \mathcal{K} \subseteq \mathbb{R}^k$ and $s \geq 0$. The idea of asking this query comes from the fact that $\widetilde{\mathrm{Fp}}(B) = \mathrm{Ball}(B^t) \oplus \widetilde{\mathrm{Fp}}(B^r)$ where by Lemma A.4.1 in Chapter 2, $\mathrm{Sep}(f, \widetilde{\mathrm{Fp}}(B)) > 0$ is equivalent with $\mathrm{Sep}(f \oplus \{-\mathrm{m}(B^t)\}, \widetilde{\mathrm{Fp}}(B^r)) > \mathrm{r}(B)$. A solution to this query is the boundary reduction method which comes from the idea of Lagrange Multiplier [11].

Given two algebraic sets $\overline{\mathcal{I}}$ and $\overline{\mathcal{K}}$ in $\mathcal{Z}$, the **extreme pair** between $\overline{\mathcal{I}}$ and $\overline{\mathcal{K}}$ is a set of pairs $(\mathbf{p}, \mathbf{q}) \in \overline{\mathcal{I}} \times \overline{\mathcal{K}}$ where each pair in the set is a local minima or maxima for the function $d_{\mathcal{Z}}(\mathbf{p}, \mathbf{q})$, denoted by $\mathrm{ext}(\overline{\mathcal{I}}, \overline{\mathcal{K}})$. We say $(\overline{\mathcal{I}}, \overline{\mathcal{K}})$ is **degenerate**, if $\mathrm{ext}(\overline{\mathcal{I}}, \overline{\mathcal{K}})$ is not a finite set. The construction for $\mathrm{ext}(\overline{\mathcal{I}}, \overline{\mathcal{K}})$ is described in Lemma 5.2.1.

Lemma 5.2.1. *Let $\mathcal{Z} = \mathbb{R}^k$. Suppose that $f_i(\mathbf{x}) = 0$ for $i = 1, \ldots, m$ are constraints for algebraic set $\overline{\mathcal{I}} \subseteq \mathcal{Z}$ and $g_j(\mathbf{x}) = (0)$ for $j = 1, \ldots, n$ are constraints for algebraic set $\overline{\mathcal{K}} \subseteq \mathcal{Z}$.*

$$\mathit{ext}(\overline{\mathcal{I}}, \overline{\mathcal{K}}) = \left\{ (\mathbf{p}, \mathbf{q}) \in \overline{\mathcal{I}} \times \overline{\mathcal{K}} : \mathbf{p} - \mathbf{q} = \sum_i \lambda_i \nabla f_i(\mathbf{p}) = \sum_j \nu_j g_j(\mathbf{q}), \exists \lambda_i, \nu_j \in \mathbb{R} \right\}$$

Proof. We use Lagrange multipliers to find the local optimas for $d_{\mathcal{Z}}(\mathbf{p}, \mathbf{q})$, which are also the local optimas for $d_{\mathcal{Z}}(\mathbf{p}, \mathbf{q})^2$. We define the Lagrangian to be

$$\mathcal{L}(\mathbf{p}, \mathbf{q}, \lambda, \nu) = d_{\mathcal{Z}}(\mathbf{p}, \mathbf{q})^2 - 2 \sum_i \lambda_i f_i(\mathbf{p}) + 2 \sum_j \nu_j g_j(\mathbf{q})$$

for some multipliers $\lambda = (\lambda_i)$ and $\nu = (\nu_j)$. The optimas are the pairs such that $\nabla_{\mathbf{p}}\mathcal{L} = \nabla_{\mathbf{q}}\mathcal{L} = \mathbf{0}$. So

$$\mathbf{0} = \nabla_{\mathbf{p}}\mathcal{L} = 2(\mathbf{p}-\mathbf{q}) - 2\sum_i \lambda_i \nabla f_i(\mathbf{p}),$$

and

$$\mathbf{0} = \nabla_{\mathbf{q}}\mathcal{L} = 2(\mathbf{q}-\mathbf{p}) + 2\sum_i \nu_i \nabla g_i(\mathbf{q}),$$

which proves the lemma. □

When $\overline{\mathcal{I}}$ and $\overline{\mathcal{K}}$ are 1-dimensional curves or 2-dimensional surfaces, we have explicit expressions for $\text{ext}(\overline{\mathcal{I}}, \mathcal{K})$, see corollaries 5.2.2 and 5.2.3. Their proofs are obvious.

Corollary 5.2.2. *Suppose that a tangent vector for a curve $\overline{\mathcal{K}}$ at $\mathbf{q} \in \overline{\mathcal{K}}$ is $\mathbf{t_q}$, then $\forall(\mathbf{p},\mathbf{q}) \in ext(\overline{\mathcal{I}}, \overline{\mathcal{K}})$, $(\mathbf{p}-\mathbf{q}) \perp \mathbf{t_q}$.*

Corollary 5.2.3. *Suppose that the normal vector for a surface $\overline{\mathcal{I}}$ at $\mathbf{p} \in \overline{\mathcal{I}}$ is $\mathbf{n_p}$, then $\forall(\mathbf{p},\mathbf{q}) \in ext(\overline{\mathcal{I}}, \overline{\mathcal{K}})$, $(\mathbf{p}-\mathbf{q}) /\!\!/ \mathbf{n_p}$.*

Lemma 5.2.1 is applicable for $\mathcal{Z} = \mathbb{R}^k$ for all k and any algebraic sets $\overline{\mathcal{I}}$ and $\overline{\mathcal{K}}$. Construction of extreme pairs is the process of finding algebraic roots for the equation system 5.1:

$$\begin{aligned} & f_i(\mathbf{p}) = 0 \\ & g_j(\mathbf{q}) = 0 \\ & (\mathbf{p}-\mathbf{q}) + \sum_i \lambda_i \nabla f_i(\mathbf{p}) = 0 \\ & (\mathbf{p}-\mathbf{q}) + \sum_j \nu_j \nabla g_j(\mathbf{q}) = 0 \end{aligned} \tag{5.1}$$

This is actually a "Zero Problem". However, SSS planner only finds roots when $\mathcal{I}$ and $\mathcal{K}$ are spans for elementary sets, where for all equations above in the system, their degrees are at most 2 and their dimensions are 3. Solving such a set of equations with bounded degrees and dimensions is $O(1)$ for inputs of different coefficients, which is implementable.

For any semi-algebraic sets $\mathcal{I}$, we denote its interior by $\mathcal{I}^\circ$. The technique of boundary reduction is based on the Theorem 5.2.4.

Theorem 5.2.4. *Let $\mathcal{I}$ and $\mathcal{K}$ be two compact semi-algebraic sets in $\mathcal{Z} = \mathbb{R}^k$. Then Sep($\mathcal{I}, \mathcal{K}$) $> s$ for some $s \geq 0$, if and only if*

1. $\forall(\mathbf{p}, \mathbf{q}) \in ext(\overline{\mathcal{I}}, \overline{\mathcal{K}}) \cap (\mathcal{I} \times \mathcal{K})$, $d_{\mathcal{Z}}(\mathbf{p}, \mathbf{q}) > s$;

2. *Sep*$(\partial\mathcal{I}, \mathcal{K}) > s$;

3. *Sep*$(\mathcal{I}, \partial\mathcal{K}) > s$.

Proof. Sep$(\mathcal{I}, \mathcal{K}) > s$ always implies the others, so we only show the sufficiency.

Since $\mathcal{I}$ and $\mathcal{K}$ are compact, there are $\mathbf{p}_0, \mathbf{q}_0 \in \mathcal{I} \times \mathcal{K}$, such that $d_{\mathcal{Z}}(\mathbf{p}_0, \mathbf{q}_0) = \text{Sep}(\mathcal{I}, \mathcal{K})$. If $\mathbf{p}_0 \in \partial\mathcal{I}$ or $\mathbf{q}_0 \in \partial\mathcal{K}$, then we know $d_{\mathcal{Z}}(\mathbf{p}_0, \mathbf{q}_0) > s$ since $\text{Sep}(\partial\mathcal{I}, \mathcal{K}) > s$ and $\text{Sep}(\mathcal{I}, \partial\mathcal{K}) > s$. Now suppose that $\mathbf{p}_0 \in \mathcal{I}^\circ$ and $\mathbf{q}_0 \in \mathcal{K}^\circ$. We show that this pair $(\mathbf{p}_0, \mathbf{q}_0) \in \text{ext}(\overline{\mathcal{I}}, \overline{\mathcal{K}})$. Since $(\mathbf{p}_0, \mathbf{q}_0) \in (\mathcal{I} \times \mathcal{K})^\circ$, there is an open neighborhood $\mathcal{U}_{\mathbf{p}_0\mathbf{q}_0} \subseteq \mathcal{I}^\circ \times \mathcal{K}^\circ$ such that it is a local minima for function $d_{\mathcal{Z}}(\mathbf{p}, \mathbf{q})$ in $\mathcal{U}_{\mathbf{p}_0\mathbf{q}_0}$. Moreover $\mathcal{U}_{\mathbf{p}_0\mathbf{q}_0} \subseteq \overline{\mathcal{I}} \times \overline{\mathcal{K}}$ which implies $(\mathbf{p}_0, \mathbf{q}_0)$ is also a local minima for function $d_{\mathcal{Z}}(\mathbf{p}, \mathbf{q})$ in $\overline{\mathcal{I}} \times \overline{\mathcal{K}}$. Then by definition, $(\mathbf{p}_0, \mathbf{q}_0) \in \text{ext}(\overline{\mathcal{I}}, \overline{\mathcal{K}})$. $\square$

Corollary 5.2.5. *Let $\mathcal{I}$ and $\mathcal{K}$ be two compact semi-algebraic sets in $\mathcal{Z} = \mathbb{R}^k$. Then Sep$(\mathcal{I}^\circ, \mathcal{K}^\circ) > s$ for some $s \geq 0$, if and only if $\forall(\mathbf{p}, \mathbf{q}) \in ext(\overline{\mathcal{I}}, \overline{\mathcal{K}}) \cap (\mathcal{I} \times \mathcal{K})$, $d_{\mathcal{Z}}(\mathbf{p}, \mathbf{q}) > s$.*

Based on the Theorem 5.2.4, the process of querying if $\text{Sep}(f, \widetilde{\text{Fp}}(B))$ is clear. The boundary of any compact semi-algebraic sets is either an empty set or a finite union of semi-algebraic sets [2]. One can begin with asking an initial query if $\text{Sep}(f, \widetilde{\text{Fp}}(B) = \cup_i \mathcal{J}_i) > 0$, and for each $\mathcal{J}_i$, recursively asking those queries of boundary primitives. Each recursion will always reduce the dimension of primitives in the queries until a point or an empty set. Note that the separation from an empty set to any other sets is always $+\infty$. The separation between two points is the distance, which is trivial to compute. Hence this reduction is always haltable and computable for non-degenerated $(f, \widetilde{\text{Fp}}(B))$.

The next section will apply Lemma 5.2.4 to give a very efficient process of answering the query $\text{Sep}(f, \widetilde{\text{Fp}}(B))$ for the Delta robot.

5.3 Explicit Collision Detection for Delta robot

In this section, we will introduce a very special class of geometric solids, which is a key to simplify the parametric separation query for $\widetilde{\mathrm{Fp}}(B)$ of Delta robots. By a **solid**, we mean a special bounded Π_1 or Σ_2 set, where the Π_1 set may be one of the following types:

$$\text{line segments, triangle, trapezoid, right cylinder, right cone, right frustum, pyramid.} \tag{5.2}$$

By right cylinder, we mean the intersection of an infinite cylinder with two half-spaces whose bounding planes are perpendicular to the axis of the cylinder. The notion of right frustum is similar, but using a doubly-infinite cone instead of a cylinder. Thus the two "ends" of a right cylinder and a right frustum are bounded by two discs, rather than general ellipses. A right cone is a special case of a right frustum when one disc is just a single point. The pyramid is that special polyhedron Pyramid(B) for each box $B \in \square\mathcal{W}$.

A right cone can be parametrized by $\mathrm{Cone}(\mathbf{v}, \mathbf{c}, r)$, where $\mathbf{v}$ is its apex, $\mathbf{c}$ is the center of the base, and r is the radius of the base. The conic surface of the right cone is denoted by $\mathrm{tc}(\mathbf{v}, \mathbf{c}, r)$, which we call a **traffic cone**. The base disc is $\mathrm{disc}(\mathbf{c}, \mathbf{n}, r)$, where $\mathbf{n} = \mathbf{v} - \mathbf{c}$ is a normal vector to the base. The very special geometric solid is a right cone $\mathrm{Cone}(\mathbf{v}, \mathbf{c}, r)$ unioned with the unique special ball $\mathrm{Ball}(\mathbf{o}, R)$ that is tangent to $\mathrm{tc}(\mathbf{v}, \mathbf{c}, r)$, see Figure 5.7. We call this special solid an **ice-cream cone**, denoted by $\mathrm{icc}(\mathbf{v}, \mathbf{o}, R)$. The relation between parameters of the two solids (right cone and ball) in $\mathrm{icc}(\mathbf{v}, \mathbf{o}, R)$ is given by Lemma 5.3.1.

Lemma 5.3.1. *In the ice-cream cone*

$$icc(\mathbf{v}, \mathbf{o}, R) = Cone(\mathbf{v}, \mathbf{c}, r) \cup Ball(\mathbf{o}, R),$$

let $h = |\overline{\mathbf{vo}}| = |\mathbf{v} - \mathbf{o}|$, *we have*

$$r = \frac{R}{h}\sqrt{R^2 - h^2},$$

and

$$\mathbf{c} = \frac{1}{R^2}\left((R^2 - r^2)\mathbf{v} + r^2\mathbf{o}\right).$$

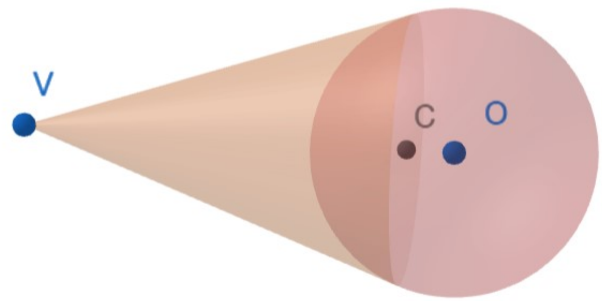


Figure 5.7: The ice-cream cone $\mathrm{icc}(\mathbf{v}, \mathbf{o}, R)$

The proof of the lemma is based on classical geometry of similarity between triangles, which is obvious.

As an application of the boundary reduction method, we decide the process of answering the parametric separation query for $\mathrm{Sep}(f, \mathrm{icc}(\mathbf{v}, \mathbf{o}, R)) > s$ by Lemma 5.3.2.

Lemma 5.3.2. *Let* $\mathbf{c}$ *and* r *be the values defined by Lemma 5.3.1.* $Sep(f, icc(\mathbf{v}, \mathbf{o}, R)) > s$ *for some* $s > 0$ *if and only if*

$$Sep(f, \{\mathbf{v}\}) > s \ \wedge \ Sep(f, tc(\mathbf{v}, \mathbf{c}, r)^\circ) > s \ \wedge \ Sep(f, Ball(\mathbf{o}, R)) > s \ \wedge \ Sep(f, Cone(\mathbf{v}, \mathbf{c}, r)^\circ) > 0.$$

Proof. The necessity is obvious, we show the sufficiency.

By reduction to boundaries,

$$\begin{aligned}
\mathrm{icc}(\mathbf{v}, \mathbf{o}, R) &= \mathrm{Cone}(\mathbf{v}, \mathbf{c}, r) \cup \mathrm{Ball}(\mathbf{o}, R) \\
&= \partial\mathrm{Cone}(\mathbf{v}, \mathbf{c}, r) \cup \mathrm{Ball}(\mathbf{o}, R) \cup \mathrm{Cone}(\mathbf{v}, \mathbf{c}, r)^\circ \\
&= (\{\mathbf{v}\} \cup \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ \cup \mathrm{disc}(\mathbf{c}, \mathbf{v} - \mathbf{c}, r)) \cup \mathrm{Ball}(\mathbf{o}, R) \cup \mathrm{Cone}(\mathbf{v}, \mathbf{c}, r)^\circ \\
&= \{\mathbf{v}\} \cup \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ \cup \mathrm{Ball}(\mathbf{o}, R) \cup \mathrm{Cone}(\mathbf{v}, \mathbf{c}, r)^\circ
\end{aligned}$$

The last equality is true since $\mathrm{disc}(\mathbf{c}, \mathbf{v} - \mathbf{c}, r) \subseteq \mathrm{Ball}(\mathbf{o}, R)$. The fourth predicate only checks if $\mathrm{Sep}(f, \mathrm{Cone}(\mathbf{v}, \mathbf{c}, r)^\circ) > 0$ since $\overline{\mathrm{Cone}(\mathbf{v}, \mathbf{c}, r)} = \mathbb{R}^3$ where distances of minima pairs are always 0. $\square$

Note that the pair in the fourth query is degenerate. But we can avoid computing infinite pairs by only checking if for an arbitrary $\mathbf{p} \in f$, $\mathbf{p} \notin \mathrm{Cone}(\mathbf{v}, \mathbf{c}, r)$, based on the success of the second query. Based on Lemma 5.3.2, the reduction process for an ice-cream cone is the following:

Query Sep$(f, \mathbf{icc}(\mathbf{v}, \mathbf{o}, R)) > s$?

Input: $\mathbf{v}, \mathbf{o} \in \mathbb{R}^3$, feature $f \subseteq \mathbb{R}^3$, $R > 0$, $s \geq 0$.

Output: `true` if $\mathrm{Sep}(f, \mathrm{icc}(\mathbf{v}, \mathbf{o}, R)) > s$, `false` else.

If $\mathrm{Sep}(f, \{\mathbf{v}\}) > s$ is `false`, return `false`.

If $\mathrm{Sep}(f, \mathrm{Ball}(\mathbf{o}, R)) > s$ is `false`, return `false`.

If $\mathrm{Sep}(f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s$ is `false`, return `false`.

Choose a point $\mathbf{p} \in f$,

if $\mathbf{p} \in \mathrm{Cone}(\mathbf{v}, \mathbf{c}, r)^\circ$, return `false`;

else return `true`.

In [23], the values of $\mathrm{Sep}(f, \mathcal{J})$ for any features f and linear solids $\mathcal{J}$ or a ball already have an explicit computing process. What's not included is detecting if $\mathrm{Sep}(f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s$. We develop these techniques. See Lemma D.3.1, Example D.3.2 and Lemma D.3.3 and their corresponding algorithms in the appendix.

The power of solid is that we can answer the query $\mathrm{Sep}(f, \widetilde{\mathrm{Fp}}(B)) > 0$ by 4 simple sub-queries. We summarize the result into a Theorem 5.3.3.

Theorem 5.3.3. *For any box $B = B^t \times B^r \in \square\mathcal{W}$ and feature f, Sep$(f, \widetilde{\mathrm{Fp}}(B)) > 0$ if and only if*

$$(Sep(icc_{\mathcal{A}}(B), f) > r(B)) \wedge (Sep(icc_{\mathcal{B}}(B), f) > r(B)) \\ \wedge \left(Sep(\overline{m_{\mathcal{A}}(B) m_{\mathcal{B}}(B)}, f) > d(B) \right)$$

$$\wedge\,(Sep(Pyramid(B), f) > 0)$$

where $icc_{\mathcal{A}}(B) = icc(m(B^t), m_{\mathcal{A}}(B), d(B^r))$ *and* $icc_{\mathcal{B}}(B) = icc(m(B^t), m_{\mathcal{B}}(B), d(B^r))$.

Proof. We rearrange our approximate footprint in this way:

$$\begin{aligned}\widetilde{\mathrm{Fp}}(B) = &(\mathrm{S}_{\mathcal{A}}(B) \cup \mathrm{Frustum}_{\mathcal{A}}(B) \cup \mathrm{S}_{\mathcal{O}}(B))\\ &\cup (\mathrm{S}_{\mathcal{B}}(B) \cup \mathrm{Frustum}_{\mathcal{B}}(B) \cup \mathrm{S}_{\mathcal{O}}(B))\\ &\cup (\mathrm{S}_{\mathcal{A}}(B) \cup \mathrm{Cylinder}(B) \cup \mathrm{S}_{\mathcal{B}}(B))\\ &\cup \mathrm{Pyramid}(B).\end{aligned}$$

Notice that:

$$\begin{aligned}\mathrm{S}_{\mathcal{A}}(B) \cup \mathrm{Frustum}_{\mathcal{A}}(B) \cup \mathrm{S}_{\mathcal{O}}(B) &= \mathrm{icc}_{\mathcal{A}}(B) \oplus \mathrm{Ball}(\mathrm{r}(B))\\ \mathrm{S}_{\mathcal{B}}(B) \cup \mathrm{Frustum}_{\mathcal{B}}(B) \cup \mathrm{S}_{\mathcal{O}}(B) &= \mathrm{icc}_{\mathcal{B}}(B) \oplus \mathrm{Ball}(\mathrm{r}(B))\\ \mathrm{S}_{\mathcal{A}}(B) \cup \mathrm{Cylinder}(B) \cup \mathrm{S}_{\mathcal{B}}(B) &= \overline{\mathrm{m}_{\mathcal{A}}(B)\mathrm{m}_{\mathcal{B}}(B)} \oplus \mathrm{Ball}(\mathrm{d}(B))\end{aligned}$$

By Lemma A.4.1, $\mathrm{Sep}(f, \widetilde{\mathrm{Fp}}(B)) > 0$ if and only if the four queries in the conjunction form described in the theorem all succeed. □

Remark 5.3.4. When $\widetilde{\mathrm{Fp}}(B^r)$ is degenerated, the ball $\mathrm{S}_{\mathcal{O}}(B) \subseteq \mathrm{Cylinder}(B)$. In this case, the solid decomposition for $\widetilde{\mathrm{Fp}}(B)$ is reduced to purely $\mathrm{Cylinder}(B)$, i.e., $\widetilde{\mathrm{Fp}}(B) = \mathrm{Cylinder}(B)$, see Figure 5.8. Therefore, the SSS planner only needs to check if $\mathrm{Sep}(\overline{\mathrm{m}_{\mathcal{A}}(B)\mathrm{m}_{\mathcal{B}}(B)}, f) > \mathrm{d}(B)$ in this degenerated case.

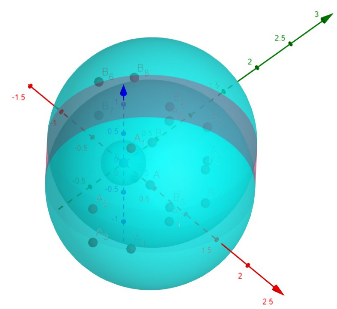


Figure 5.8: When $\mathrm{w}(B^r) \geq 1$, $\widetilde{\mathrm{Fp}}(B^r)$ is degenerated.

Chapter 6

Delta Subdivision Scheme and Box Adjacency

An SSS planner plans a path by subdividing the box space $\mathcal{W} = \mathbb{R}^7$ until finding a channel from the start to the goal box. The channel is constructed from the graph G of `FREE` boxes. The planner can use standard discrete search strategies such as Breadth First Search (BFS), the Dijkstra Algorithm [16], the A* Method [62], etc. The graph G depends on two aspects, the subdivision scheme and the data structure maintaining the subdivision. In this chapter, we will discuss the subdivision scheme for the Delta robot and design a product tree structure to maintain the subdivision.

6.1 Subdivision Scheme

Recall that the subdivision in the SSS framework is based on the technique of the well-developed interval method (also called interval arithmetic) [40]. Based on the interval method, the framework builds a subdivision tree. We will first review concepts related to subdivision trees [8].

6.1.1 Subdivision Tree

We first consider subdivision of the standard cube $B_0 = [-1, 1]^d$ in $d \geq 1$ dimensions. A **subdivision tree** $\mathcal{T}(B_0)$ is a finite tree rooted at $[-1, 1]^d$ whose nodes are subboxes of $[-1, 1]^d$. Each internal node has 2^k congruent children for some $k \in \{1, \ldots, d\}$. The set of leaves of $\mathcal{T}(B_0)$ constitute a **subdivision** of $[-1,1]^d$. Nodes of $\mathcal{T}(B_0)$ are called **aligned boxes** and every aligned box has a natural **depth**, where the root has depth 0.

Given an interval I, we denote I^- to be the negative endpoint of I and denote I^+ to be the positive endpoint of I.

Let $k = -1, 0, \ldots, d$. Two boxes B, B' are **k-adjacent** if $B \cap B'$ is an k-dimensional box. If they are d-adjacent, we say B and B' **overlap**. If they are $(d-1)$-adjacent, we say they are **neighbors**. As a matter of fact, if B and B' are overlapping aligned boxes, then either $B \subseteq B'$ or $B' \subseteq B$.

An **indicator** is an array of entries $\mathbf{d} \in \{\overline{1}, 0, 1\}^d$. For example, $(1, \overline{1}, 0)$ is an indicator for $d = 3$. A **flip** of an indicator $\mathbf{d}$ exchanges $\overline{1}$ with 1, denoted by $\text{flip}(\mathbf{d})$. For instance, $\text{flip}(1, \overline{1}, 0) = (\overline{1}, 1, 0)$.

If an indicator $\mathbf{d}$ has no $\overline{1}$, such as $(1, 0, 1, 0)$, we call it a **split indicator**. The sum of all its entries is the **order** of the split indicator, denoted by $\text{ord}(\mathbf{d})$. Given a leaf B in a subdivision $\mathcal{T}(B_0)$ and a split indicator $\mathbf{d}$, the **split** of B by split indicator $\mathbf{d}$, $\text{split}_{\mathbf{d}}(B)$, is the set of $2^{\text{ord}(\mathbf{d})}$ congruent children, such that the j-th components I_j of $B = \prod_{i=1}^d I_i$ indicated by 1 in $\mathbf{d}$ are subdivided into 2 equal subintervals. If $\text{ord}(\mathbf{d}) = d$, then we call it a **total split**. The subdivision tree $\mathcal{T}(B_0)$ is constructed by repeatedly splitting the root B_0.

If an indicator $\mathbf{d}$ has exactly one non-zero component, such as $(0, 0, 1, 0)$, we call it a **direction**; if the non-zero component is 1, it is a **positive direction**, otherwise it is a **negative direction**, as in $(0, 0, \overline{1}, 0)$. The direction with j-th component positive is $\mathbf{e}_j$ and with j-th component negative is $-\mathbf{e}_j$. The set of all directions is $\mathbf{Dir} = \{\pm\mathbf{e}_j : j = 1, \ldots, d\}$.

If an indicator $\mathbf{d}$ has no zero components, such as $(1, \overline{1}, \overline{1})$, we call it a **child indicator**. Child indicators uniquely identify each child in a total split. For example, in splitting $[-1,1]^3$, the child corresponding to $(1, \overline{1}, \overline{1})$ is $[0,1] \times [-1,0] \times [-1,0]$. We denote the child indicator of a non-root box B as $\text{childId}(B)$.

Given a box B, we can project it in one of d directions: $\text{Proj}_i(B) := \prod_{j=1, j\neq i}^d I_j$ is a $(d-1)$ dimensional box for $i \in \{1, \ldots, d\}$,. We also have a reverse process of projections, $\otimes_i$, defined as

$$B = \operatorname*{Proj}_i(B) \otimes_i I_i$$

for any box B and any $i \in \{1, \ldots, d\}$.

Any box B has k-dimensional **faces** for $k = 0, \ldots, d$. If $k = 0$, we call the face a **corner**. If $k = d-1$, we call the face a **facet**. The facet in the direction $\mathbf{e}_j$ is

$$\text{facet}_{\mathbf{e}_j}(B) = \operatorname*{Proj}_j(B) \otimes_j I_j^+$$

and the facet to the direction $-\mathbf{e}_j$ is

$$\text{facet}_{\mathbf{e}_j}(B) = \operatorname*{Proj}_j(B) \otimes_j I_j^- .$$

Given two boxes $B, B' \in \mathcal{T}(B_0)$, if B is a child of B', we write $B = \text{child}(B')$ and $B' = \text{parent}(B)$. If B and B' are $(d-1)$-adjacent, there is a unique direction $\mathbf{d}$ such that B' is **adjacent to** B in direction $\mathbf{d}$, denoted by $B \xrightarrow{\mathbf{d}} B'$. Note that if $B \xrightarrow{\mathbf{d}} B'$, then $B \cap B' \subseteq \text{facet}_{\mathbf{d}}(B)$.

Now for $B_0 = B_0^t \times \widehat{SO}(3)$ box space, the subdivision tree is $\mathcal{T}(B_0) = \mathcal{T}(B_0^t) \times \mathcal{T}(\widehat{SO}(3))$, where $\mathcal{T}(B_0^t)$ and $\mathcal{T}(\widehat{SO}(3))$ are called **template tree**. Each node B of $\mathcal{T}(B_0)$ is represented by a pair of nodes (B^t, B^r) in $\mathcal{T}(B_0^t) \times \mathcal{T}(\widehat{SO}(3))$. We will demonstrate this product operation later in the next section. Here, we describe the construction of the two templates.

In this thesis, we only make total splits on both of the two template trees $\mathcal{T}(B_0^t)$ and $\mathcal{T}(\widehat{SO}(3))$. The total splits in $\mathcal{T}(B_0^t)$ are called ***T*-splits** and the total splits in $\mathcal{T}(\widehat{SO}(3))$ are called ***R*-splits**. The template tree of $\mathcal{T}(B_0^t)$ is the same as the standard subdivision tree of $[-1, 1]^3$, which we call an **oct-tree**. The template tree of $\mathcal{T}(\widehat{SO}(3))$ differs slightly. The root of $\mathcal{T}(\widehat{SO}(3))$ is the $\widehat{SO}(3)$. Its first split divides $\widehat{SO}(3)$ into 4 children:

$$C_w = \{(-1, x, y, z) : (x, y, z) \in [-1, 1]^3\},$$

$$C_x = \{(w, -1, y, z) : (w, y, z) \in [-1, 1]^3\},$$

$$C_y = \{(w, x, -1, z) : (w, x, z) \in [-1, 1]^3\},$$

$$C_z = \{(w, x, y, -1) : (w, x, y) \in [-1, 1]^3\}.$$

The four boxes C_w, C_x, C_y, C_z are called the **subroots** of $\mathcal{T}(\widehat{SO}(3))$, and we alternatively denote them as C_1, C_2, C_3, C_4, respectively. Subsequent splits occur within one of the subroots. Each subroot forms a tree, $\mathcal{T}(C_i)$ for $i = 1, 2, 3, 4$, where each $\mathcal{T}(C_i)$ is an oct-tree, except that all indicators for each $\mathcal{T}(C_i)$, including split indicators, child indicators and directions, adds a "$*$" to the i-th component. For example, a split of a box B in $C_w = C_1 \subseteq \widehat{SO}(3)$ by a split indicator $(*, 0, 1, 0)$ results in 4 boxes whose child indicators are $(*, \overline{1}, 0, \overline{1})$, $(*, 1, 0, \overline{1})$, $(*, \overline{1}, 0, 1)$, and $(*, 1, 0, 1)$ respectively.

6.1.2 Subdivision Scheme for the Delta Robot

For simplicity, the SSS planner for Delta robot will only apply one of T-split or R-split when splitting a node of $\mathcal{T}(B_0)$, which we refer to as a TR**-subdivision scheme**, or simply a TR-scheme. Note that in previous papers [56], this term was used to refer to the process of first performing purely T-splits until $\mathrm{w}(B^t) < \varepsilon$, and then performing R-splits. The $\square\mathcal{W}$ for the Delta robot is defined as all possible boxes that may be obtained from the TR-subdivision scheme, from the initial $B_0 = B_0^t \times \widehat{SO}(3)$. In the case of TR-scheme, if T-split is applied, we have $\mathrm{w}(B^t) = \frac{1}{2}\mathrm{w}(\mathrm{parent}(B)^t)$, and if R-split is applied, we have $\mathrm{w}(B^r) = \frac{1}{2}\mathrm{w}(\mathrm{parent}(B)^r)$. Additionally, we have $\ell(B^t) = \sqrt{3}\mathrm{w}(B^t)$ and $\ell(B^r) = \sqrt{3}\mathrm{w}(B^r)$ for any box $B \in \square\mathcal{W}$.

The subdivision scheme for an SSS planner is required to guarantee the inheritance of the feature set and satisfy the Axiom **(A1)**. We discuss these two problems next.

Let us first consider the inheritance property of approximate feature sets. Recall that $\widetilde{\phi}(B)$ is the approximate feature set of B. The inheritance property is $\widetilde{\phi}(B) \subseteq \widetilde{\phi}(\mathrm{parent}(B))$, which is implied by $\widetilde{\mathrm{Fp}}(B) \subseteq \widetilde{\mathrm{Fp}}(\mathrm{parent}(B))$. Since we only do T-split or R-split each time, it is either $B^t = \mathrm{parent}(B)^t$ or $B^r = \mathrm{parent}(B)^r$. For the second case, the inheritance is obvious, since

$$\widetilde{\mathrm{Fp}}(B) = \mathrm{Ball}(B^t) \oplus \widetilde{\mathrm{Fp}}(B^r)$$

$$
\begin{aligned}
&= \mathrm{Ball}(B^t) \oplus \widetilde{\mathrm{Fp}}(\mathrm{parent}(B)^r)) \\
&\subseteq \mathrm{Ball}(\mathrm{parent}(B)^t) \oplus \widetilde{\mathrm{Fp}}(\mathrm{parent}(B)^r)) \\
&= \widetilde{\mathrm{Fp}}(\mathrm{parent}(B))
\end{aligned}
$$

For the first case, it looks true for general cases except when $\mathrm{w}(B^r) = 1$ or $\mathrm{w}(B^r) = 2$, where $\mathrm{w}(B^r) = 2$ implies $B^r = C_w$ or C_x or C_y or C_z, and $\mathrm{parent}(B)^r = \widehat{SO}(3)$. But this is easy to fix since we can define $\widetilde{\mathrm{Fp}}(B^r)$ as a larger ball when $\mathrm{w}(B^r) \geq 1$. The general case is given by a Conjecture 6.1.1.

Conjecture 6.1.1. *Let $B_1 \in \square\mathcal{W}$. If $w(B_1) \leq 1$, then for any box $B_2 \subseteq B$, $\widetilde{\mathrm{Fp}}(B_2^r) \subseteq \widetilde{\mathrm{Fp}}(B_1^r)$.*

We discovered this conjecture by experiments using Geogebra. We observe experimentally that balls $\widetilde{\mathrm{Fp}}_{\mathbf{e}_k}(B^r)$ for $k = 1, 2$ are monotonically shrinking when shrinking B for $\mathrm{w}(B) \leq 1$. We have not proved this conjecture yet. But there is a method to show that the feature set defined by collecting features only from the parent still works for the SSS framework. We will introduce this method in Example E.1.1 in the appendix.

Based on the conjecture, the $\widetilde{\mathrm{Fp}}$ satisfies the inheritance property since when $B^t = \mathrm{parent}(B)^t$ for $\mathrm{w}(B^r) \leq \frac{1}{2}$,

$$
\begin{aligned}
\widetilde{\mathrm{Fp}}(B) &= \mathrm{Ball}(B^t) \oplus \widetilde{\mathrm{Fp}}(B^r) \\
&= \mathrm{Ball}(\mathrm{parent}(B)^t) \oplus \widetilde{\mathrm{Fp}}(B^r)) \\
&\subseteq \mathrm{Ball}(\mathrm{parent}(B)^t) \oplus \widetilde{\mathrm{Fp}}(\mathrm{parent}(B)^r)) \\
&= \widetilde{\mathrm{Fp}}(\mathrm{parent}(B)).
\end{aligned}
$$

The bound D_0 in **(A1)** is achieved by bounding the aspect ratio for each B in the scheme. In fact, for purely T-split or purely R-split, aspect ratios $\alpha(B^t) = \alpha(B_0^t)$ and $\alpha(B^r) = \sqrt{3}$ are always constants. For the TR-scheme, one only needs to bound the ratio $\mathrm{w}(B^t)/\mathrm{w}(B^r)$ for each box $B = B^t \times B^r$. This depends on the details of our subdivision scheme for the Delta robot. We should decide for each box B, when to do T-split and when to do R-split. We propose the following scheme for the Delta robot:

Delta Subdivision Scheme $TR(B)$

Input: box $B \in \square\mathcal{W}$.

Output: "T-split" or "R-split".

If $\mathrm{w}(B^t) > 0.25$, return "T-split".

If $\mathrm{w}(B^r) > 0.25$, return "R-split".

If $\mathrm{w}(B^t) \geq \mathrm{w}(B^r)$, return "$T$-split".

return "R-split".

We design this scheme so that in each split, the reduction in the volume of $\widetilde{\mathrm{Fp}}(B)$ is approximately maximized. For simplicity, no matter what D_0 is, we assume that for $B_0 = \cup_{i=1}^{n} B_i$, each B_i satisfies $\alpha(B_i) \leq D_0$. Otherwise, we can preprocess B_0 by cutting the longest edges of B_i until each B_i satisfies $\alpha(B_i) \leq D_0$. Then, this special scheme TR has a bounded aspect ratio D_0 independent to input. We summarize it into Proposition 6.1.2.

Proposition 6.1.2. *Let $B_0 = dom(\mu)$. The TR scheme has a bounded aspect ratio*

$$D_0 = 9\sqrt{3}.$$

Proof. When $\mathrm{w}(B^t) > \frac{1}{4}$, it is true by assumption of the initial box. When $\mathrm{w}(B^t) \leq \frac{1}{4}$ and $\mathrm{w}(B^r) > \frac{1}{4}$, we will not do T-split, so $\mathrm{w}(B^t) \geq \frac{1}{8}$. Since $\ell(\widehat{SO}(3)) = 2\sqrt{3}$, we have

$$\alpha(B) \leq (\ell(\widehat{SO}(3)) + \ell(B^t))/\mathrm{w}(B^t) = 9\sqrt{3}.$$

When $\mathrm{w}(B^t) \leq \frac{1}{4}$ and $\mathrm{w}(B^r) \leq \frac{1}{4}$. If $\mathrm{w}(B^t) \geq \mathrm{w}(B^r)$, we will not do R-split, so $\mathrm{w}(B^r) \geq \frac{1}{2}\mathrm{w}(B^t)$. Then $\alpha(B) \leq (\ell(B^t) + \ell(B^r))/\mathrm{w}(B^r) \leq 3\sqrt{3}$. The case when $\mathrm{w}(B^t) \leq \frac{1}{4}$ and $\mathrm{w}(B^r) \leq \frac{1}{4}$ but $\mathrm{w}(B^t) < \mathrm{w}(B^r)$ is the same. Therefore, $\alpha(B) \leq 9\sqrt{3}$ for any $B \in \square\mathcal{W}$. $\square$

Corollary 6.1.3. *The maximum aspect ratio for ε-small boxes in the TR scheme is $2\sqrt{3}$ when $\varepsilon < 0.125$.*

6.2 Data Structure Maintaining Adjacency of Boxes

We have discussed the subdivision scheme for the Delta robot and showed that it has the inheritance property. The aspect ratio is bounded by $9\sqrt{3}$. The other problem is to maintain the adjacency of boxes in the subdivision tree. This process requires quickly finding adjacent boxes given a new `FREE` box B, and it is based on the data structure of subdivision trees.

To describe the method of finding neighbors, we introduce the concepts of reverse direction, **d**-cousin, and principal **d**-neighbor.

6.2.1 Reverse of Direction

We use $\mathcal{T}(B_0^*)$ to denote either $\mathcal{T}(B_0^t)$ or $\mathcal{T}(\widehat{SO}(3))$. Given a box B in the $\mathcal{T}(B_0^*)$ and $\mathbf{d} \in \mathbf{Dir}$, the **reverse direction** of $\mathbf{d}$ is $\mathrm{rev}_B(\mathbf{d}) \in \mathbf{Dir}$ such that if $B \xrightarrow{\mathbf{d}} B'$, then $B' \xrightarrow{\mathrm{rev}_B(\mathbf{d})} B$. The reverse direction is well defined according to Theorem 6.2.1, since we can define $\mathrm{rev}_B(\mathbf{d})$ by the unique smallest-depth box B' such that $B \xrightarrow{\mathbf{d}} B'$.

Theorem 6.2.1. *$rev_{B_2}(\mathbf{d}) = rev_{B_1}(\mathbf{d})$ for any B_2 that in $\mathcal{T}(B_1)$ that is a boundary of B_1 in direction $\mathbf{d} \in \boldsymbol{Dir}$.*

Proof. This is because B_1 and B_2 share the same facet, that is, $\mathrm{facet}_{\mathbf{d}}(B_2) \subseteq \mathrm{facet}_{\mathbf{d}}(B_1)$. For any B_3 such that $B_1 \xrightarrow{\mathbf{d}} B_3$ and B_4 such that $B_2 \xrightarrow{\mathbf{d}} B_4$,

$$B_2 \cap B_4 \subseteq \mathrm{facet}_{\mathrm{rev}_{B_2}(\mathbf{d})}(B_4),$$

and

$$B_2 \cap B_4 \subseteq B_1 \cap B_3 \subseteq \mathrm{facet}_{\mathrm{rev}_{B_1}(\mathbf{d})}(B_3).$$

Then $\mathrm{facet}_{\mathrm{rev}_{B_2}(\mathbf{d})}(B_4) \cap \mathrm{facet}_{\mathrm{rev}_{B_1}(\mathbf{d})}(B_3) \neq \emptyset$ implies $\mathrm{rev}_{B_2}(\mathbf{d}) = \mathrm{rev}_{B_1}(\mathbf{d})$. □

In fact, if $B \xrightarrow{\mathbf{d}} B'$, then either $\mathrm{facet}_{\mathbf{d}}(B) \subseteq \mathrm{facet}_{\mathrm{rev}_B(\mathbf{d})}(B')$ or $\mathrm{facet}_{\mathbf{d}}(B) \supseteq \mathrm{facet}_{\mathrm{rev}_B(\mathbf{d})}(B')$.

In $\mathcal{T}(B_0^t)$, the direction $\mathrm{rev}_{B^t}(\mathbf{d})$ is $-\mathbf{d}$ for any box B^t. Similar result for B^r in $\widehat{SO}(3)$ tree if B^r is not a boundary of C_i for any $i \in \{w, x, y, z\}$, since it is locally $\mathbb{R}^3$ around facet $\mathrm{facet}_{\mathbf{d}}(B^r)$. For B^r that is a boundary of C_i, $\mathrm{rev}_{B^r}(\mathbf{d})$ is given by Proposition 6.2.2 and Corollary 6.2.3.

Proposition 6.2.2. *Given an oct-tree $\mathcal{T}(\widehat{SO}(3))$, for any $i, j \in \{1, 2, 3, 4\}$, $i \neq j$, $rev_{C_i}(\mathbf{e}_j) = \mathbf{e}_i$ and $rev_{C_i}(-\mathbf{e}_j) = -\mathbf{e}_i$.*

Proof. Since

$$
\begin{aligned}
\mathrm{facet}_{\mathbf{e}_j}(C_i) &= \mathrm{facet}_{\mathbf{e}_j}\left([-1,1]^3 \otimes_i \{-1\}\right) \\
&= [-1,1]^2 \otimes_i \{-1\} \otimes_j \{1\} \\
&= \mathrm{facet}_{-\mathbf{e}_i}\left([-1,1]^3 \otimes_j \{1\}\right) \\
&= \mathrm{facet}_{-\mathbf{e}_i}\left([1,-1]^3 \otimes_j \{-1\}\right) \\
&= \mathrm{facet}_{\mathbf{e}_i}\left([-1,1]^3 \otimes_j \{-1\}\right) \\
&= \mathrm{facet}_{\mathbf{e}_i}(C_j),
\end{aligned}
$$

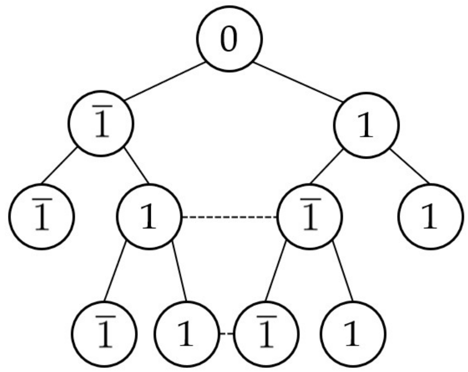
we have $\mathrm{rev}_{C_i}(\mathbf{e}_j) = \mathbf{e}_i$. The other case is similar. $\square$

Corollary 6.2.3. *Given an oct-tree $\mathcal{T}(\widehat{SO}(3))$, for any $i, j \in \{1, 2, 3, 4\}$, $i \neq j$, $rev_{B^r}(\mathbf{e}_j) = \mathbf{e}_i$ for boundary $B^r \subseteq C_i$ in direction $\mathbf{e}_j$.*

6.2.2 d-neighbor

Recall that a **d**-neighbor of a box B is a box B' such that $B \xrightarrow{\mathbf{d}} B'$. Given $\mathcal{T}(B_0^*)$, box B' is called a **d-cousin** of box B if B' is a **d**-neighbor of B such that $\mathrm{depth}(B) = \mathrm{depth}(B')$. In this case, $B \cap B' = \mathrm{facet}_{\mathbf{d}}(B)$. Since the depth of the cousins are the same, the width of a box is the same with its cousins. If B' is a **d**-cousin of B, then it is always the unique **d**-cousin, denoted by $\mathrm{cousin}_{\mathbf{d}}(B)$. Note that $\mathrm{cousin}_{\mathbf{d}}(B) = \texttt{NULL}$ if it is not a boundary in direction **d** or it is a root or the root or subroot of B has no **d**-neighbor or all its **d**-neighbors are "piblings". By a similar convention, a **d-sibling** B' of B means that B' is a sibling of B and $B \xrightarrow{\mathbf{d}} B'$, denoted by $B' = \text{sibling}_{\mathbf{d}}(B)$. Note that $\text{sibling}_{\mathbf{d}}(B) = \texttt{NULL}$ if B is a boundary of its parent in direction $\mathbf{d}$ or it is a root or a subroot. Since the depths between siblings are always the same, the $\text{sibling}_{\mathbf{d}}(B)$ is unique and $\text{sibling}_{\mathbf{d}}$ is well defined.

There are relations between child indicators of a box B with its $\mathbf{d}$-cousin.



Theorem 6.2.4. *Assume that the splits in the subdivision tree are all total splits. Let B_2 be a child of B_1. If childId(B_1) = childId(B_2), then*

$$\textit{childId}(\textit{cousin}_{\mathbf{d}}(B_1)) = \textit{childId}(\textit{cousin}_{\mathbf{d}}(B_2))$$

provided that $\textit{cousin}_{\mathbf{d}}(B_1)$ and $\textit{cousin}_{\mathbf{d}}(B_2)$ are not `NULL`*.*

Proof. Let $\mathbf{d} = \mathbf{e}_k$, $B_3 = \text{cousin}_{\mathbf{e}_k}(B_1)$, and $B_4 = \text{cousin}_{\mathbf{e}_k}(B_2)$. We have assumed that the splits are total, therefore the indicators of splits for B_1 and B_2 are the same, so as B_3 and B_4. Let $\mathbf{t}_j$ be the indicator of B_j in its split for $j = 1, 2, 3, 4$. Then $\text{childId}(B_1) = \text{childId}(B_2)$ implies $\mathbf{t}_1 = \mathbf{t}_2$. Since $\text{cousin}_{\mathbf{e}_k}$ exists for B_1 and B_2, the $\mathbf{e}_k$-neighbors of B_1 and B_2 cannot be their siblings, hence $\text{facet}_{\mathbf{e}_k}(B_2) \subseteq \text{facet}_{\mathbf{e}_k}(B_1)$. Let $\mathbf{t}'_i$ be the indicator of $\text{facet}_{\mathbf{e}_k}(B_i)$. Then since $\text{facet}_{\mathbf{e}_k}(B_i) = \text{Proj}_{\mathbf{e}_k}(B_i) \otimes_k \{1\}$ for positive $\mathbf{e}_k$ and $\text{facet}_{\mathbf{e}_k}(B_i) = \text{Proj}_k(B_i) \otimes_k \{-1\}$ for negative $\mathbf{e}_k$, we always have $\text{Proj}_k(\mathbf{t}_i) = \mathbf{t}'_i$. It implies that $\mathbf{t}'_1 = \mathbf{t}'_2$.

As $B_3 = \text{cousin}_{\mathbf{e}_k}(B_1)$, and $B_4 = \text{cousin}_{\mathbf{e}_k}(B_2)$, we have $\text{facet}_{\mathbf{e}_k}(B_1) = \text{facet}_{\text{rev}_{B_1}(\mathbf{e}_k)}(B_3)$ and $\text{facet}_{\mathbf{e}_k}(B_2) = \text{facet}_{\text{rev}_{B_2}(\mathbf{e}_k)}(B_4)$. Therefore $\mathbf{t}'_1 = \mathbf{t}'_3$ and $\mathbf{t}'_2 = \mathbf{t}'_4$. Since B_2 is a boundary of B_1 in direction $\mathbf{d}$, by Theorem 6.2.1, $\text{rev}_{B_1}(\mathbf{e}_k) = \text{rev}_{B_2}(\mathbf{e}_k)$. So if $\text{rev}_{B_1}(\mathbf{e}_k)$ is positive, then

$$\mathbf{t}_3 = \mathbf{t}'_3 \otimes_k \{1\} = \mathbf{t}'_1 \otimes_k \{1\} = \mathbf{t}'_2 \otimes_k \{1\} = \mathbf{t}'_4 \otimes_k \{1\} = \mathbf{t}_4,$$

and similar when $\text{rev}_{B_1}(\mathbf{e}_k)$ is negative. Therefore, $\mathbf{t}_3 = \mathbf{t}_4$, which implies $\text{childId}(B_3) = \text{childId}(B_4)$, since the indicators of splits for B_3 and B_4 are the same. $\square$

According to Theorem 6.2.4, an SSS plannar only need to decide the $\text{childId}(\text{cousin}_{\mathbf{d}}(B))$ according to $\text{childId}(B)$ for B is a root or a subroot. Let us discuss the relation between $\text{childId}(\text{cousin}_{\mathbf{d}}(B))$ and $\text{childId}(B)$ for roots and subroots B in $\mathcal{T}(B_0^t)$ and $\mathcal{T}(\widehat{SO}(3))$.

Proposition 6.2.5. *For boxes B in $\mathcal{T}(B_0^t)$,*

$$\textit{childId}(\textit{cousin}_{\pm\mathbf{e}_j}(B)) = \textit{flip}_j(\textit{childId}(B)).$$

One can show that by enumerating the first split of B_0^t. It is also true for B in $\mathcal{T}(\widehat{SO}(3))$ if B is not a boundary of C_j for $j = 1, 2, 3, 4$, since it is locally $\mathbb{R}^3$ within each C_j. For B is a boundary of C_j, we have another formula given by the flip operation. Note that, in relation between childId for different subroots in $\mathcal{T}(\widehat{SO}(3))$, $*$ is counted as $\bar{1}$ since our boxes C_i fix i-th component as -1.

Proposition 6.2.6. *Assume all splits are total splits. For B in $\mathcal{T}(\widehat{SO}(3))$ such that B is a boundary of C_j in direction $\mathbf{e}_j$,*

$$\textit{childId}(\textit{cousin}_{\mathbf{e}_j}(B)) = \textit{flip}(\textit{childId}(B)).$$

For B in $\mathcal{T}(\widehat{SO}(3))$ such that B is a boundary of C_j in direction $\mathbf{e}_j$,

$$\textit{childId}(\textit{cousin}_{-\mathbf{e}_j}(B)) = \textit{childId}(B).$$

Proof. We call the box that is a child of a subroot in $\mathcal{T}(\widehat{SO}(3))$ a **Family Patriarch** (FP). The key point is that any two FPs from different subroots in $\mathcal{T}(\widehat{SO}(3))$ exactly share a unique same corner among the 8 corners of each subroots. The child indicator for each FP is a substitution from (w, x, y, z) of a corner to $\{\bar{1}, 1, *\}^4$. For an FP box B in C_i, $\text{childId}(B)_i = *$. If B shares corner $\mathbf{b}$, then if $b_i = -1$, $\text{childId}(B)_j = \begin{cases} 1 & b_j = 1 \\ \bar{1} & b_j = -1 \end{cases}$ and if $B_i = 1$, $\text{childId}(B)_j = \begin{cases} \bar{1} & b_j = 1 \\ 1 & b_j = -1 \end{cases}$. Boxes to the negative directions will move to a negative FP in the subroot, so its childId will not change except a substitution between $*$ and $\bar{1}$. Boxes to the positive directions will move to a positive FP in the subroot, so its childId will flip with a substitution between $*$ and 1. These observations prove the proposition for FP boxes. The proposition is true by Theorem 6.2.4. $\square$

To give out an explicit view, the childId relations between any pair of adjacent FPs B_1 and B_2 are given in a table in Table E.2.1 in appendix.

In principle, the techniques for child indicators are already enough for determining the **d**-cousin of boundary boxes. But we have a much more powerful data structure that can trace tree-path indicators for all boxes with very few space occupancy. We depict this data structure in Example E.3.1 in the appendix.

6.2.3 Principal d-neighbor

The key point to quickly suggest neighbors is the technique of principal neighbors. This idea was first used in [21]. Given a subdivision tree $\mathcal{T}(B_0)$, a **principal d-neighbor** $\mathrm{princ}_{\mathbf{d}}(B)$ of box B is a **d**-neighbor of box B that is as small as possible, such that each leaf in $\mathcal{T}(B_0)$ that is a **d**-neighbor of B is either $\mathrm{princ}_{\mathbf{d}}(B)$ or a child of $\mathrm{princ}_{\mathbf{d}}(B)$. For each box B in $\mathcal{T}(B_0)$, its principal **d**-neighbor is encoded as a pointer to a box in $\mathcal{T}(B_0)$. Each box $B \in \mathcal{T}(B_0)$ has $2d$ pointers points to $\pm\mathbf{e}_j$-principal neighbors for $j = 1, \ldots, d$.

Given a subdivision tree $\mathcal{T}(B_0)$, there is a very natural strategy to assign a principal **d**-neighbor for any subdivision tree, see the process below:

Principal Neighbor $\mathbf{princ_d}(B)$

Input: box B in $\mathcal{T}(B_0)$, direction $\mathbf{d} = \pm\mathbf{e}_j$ for some j.

Output: $\text{princ}_{\mathbf{d}}(B)$.

If $B = B_0$, return `NULL`.

If B is a subroot, return $\text{cousin}_{\mathbf{d}}(B)$.

If $\text{sibling}_{\mathbf{d}}(B) \neq$ `NULL`, return $\text{sibling}_{\mathbf{d}}(B)$.

If $\text{cousin}_{\mathbf{d}}(B) \neq$ `NULL`, return $\text{cousin}_{\mathbf{d}}(B)$.

return $\text{princ}_{\mathbf{d}}(\text{parent}(B))$.

Note that this process can be dynamically programmed by recording pointers of $\text{princ}_{\mathbf{d}}$ so that during each expansion, $\text{princ}_{\mathbf{d}}(\text{parent}(B))$ is just called from the pointer from the $\text{parent}(B)$. The $\text{princ}_{\mathbf{d}}(B)$ defined by the process is correct by Lemma 6.2.7.

Lemma 6.2.7. *Any leaf* $\mathbf{d}$*-neighbor of a box* B *is a boundary of* $\mathit{princ}_{\mathbf{d}}(B)$ *in direction* $\mathit{rev}_{\mathbf{d}}(B)$ *defined in the process "Principal Neighbor".*

Proof. When B is the root or a subroot, it is obvious. We consider B' be any $\mathbf{d}$-neighbor of B such that $\text{depth}(B') \geq \text{depth}(B)$. If $\text{sibling}_{\mathbf{d}}(B) \neq$ `NULL`, B is not a $\mathbf{d}$-boundary child, so $\text{facet}_{\mathbf{d}}(B) = \text{facet}_{\text{rev}_B(\mathbf{d})}(\text{sibling}_{\mathbf{d}}(B)) \supseteq \text{facet}_{\text{rev}_B(\mathbf{d})}(B')$ which implies B' is a descendant of $\text{sibling}_{\mathbf{d}}(B)$. Similar when $\text{cousin}_{\mathbf{d}}(B) \neq$ `NULL`, under after this case, B is always a $\mathbf{d}$-boundary child of $\text{parent}(B)$. In this case, if $\text{princ}_{\mathbf{d}}(\text{parent}(B))$ statiesfies this Lemma 6.2.7, then $\text{facet}_{\text{rev}_B(\mathbf{d})}(B') \supseteq \text{facet}_{\mathbf{d}}(B) \supseteq \text{facet}_{\mathbf{d}}(\text{parent}(B))$ implies B' is a descendant of $\text{princ}_{\mathbf{d}}(\text{parent}(B))$. $\square$

For any subdivision tree with purely total splits, the process "Principal Neighbor" defines $\text{princ}_{\mathbf{d}}(B)$ for each box B by a $\mathbf{d}$-neighbor B' such that $\text{depth}(B)$ is maximized restricted to $\text{depth}(B') \leq \text{depth}(B)$. This applies for the oct-trees for both $\mathcal{T}(B_0^t)$ and $\mathcal{T}(\widehat{SO}(3))$.

6.2.4 Product Tree

Fix two template trees $\mathcal{T}(B_0^t)$ and $\mathcal{T}(\widehat{SO}(3))$. To preserve adjacency between any two boxes in B_0, we combine them into a product tree $\mathcal{T}(B_0) = \mathcal{T}(B_0^t) \times \mathcal{T}(\widehat{SO}(3))$, where each node in $\mathcal{T}(B_0)$ contains two pointers which points to two nodes, one in $\mathcal{T}(B_0^t)$ and one in $\mathcal{T}(\widehat{SO}(3))$. Initially, $\mathcal{T}(B_0^t)$ and $\mathcal{T}(\widehat{SO}(3))$ are trivial trees rooted at B_0^t and $\widehat{SO}(3)$ respectively. They grow as splits occur in $\mathcal{T}(B_0)$. When performing T-split on a box $B = B^t \times B^r$, we first split B^t in $\mathcal{T}(B_0^t)$ if B^t is a leaf of $\mathcal{T}(B_0^t)$, and then construct children of B by assigning the appropriate products of B^t and B^r. Similar operations for R-split.

The directions in B_0 are coded by directions in $\mathbb{R}^d$ for $d = 7$. $\pm\mathbf{e}_j$ is classified as a direction in B_0^t if $j \leq 3$, and is classified as a direction in $\widehat{SO}(3)$ if $j \geq 4$. For simplicity, given a direction $\mathbf{d}$, we say it is a T-direction for a direction in B_0^t or it is an R-direction otherwise.

The child indicator for each box B in $\mathcal{T}(B_0)$ is a pair $(\text{childId}(B^t), \text{childId}(B^r))$. The $\mathbf{d}$-sibling and $\mathbf{d}$-cousins in $\mathcal{T}(B_0)$ are defined by a similar process. A box B is a $\mathbf{d}$-root if B^t is a root when $\mathbf{d}$ is a T-direction or B^r is a root when $\mathbf{d}$ is an T-direction. A box B is a $\mathbf{d}$-subroot if B^r is a subroot and $\mathbf{d}$ is an T-direction.

Based on root, subroot, sibling and cousin, the principal neighbor for an $\widehat{SE}(3)$ box is defined by the "Principal Neighbor", and we have proved that it can correctly find possible neighbors based on Lemma 6.2.7. The explicit process for finding all possible neighbors is the following:

Find all neighbors of a box B in $\mathcal{T}(B_0)$: AllNeighbor(B)

Input: box $B \in \mathcal{T}(B_0)$.

Output: the set of all neighbors of B in $\mathcal{T}(B_0)$.

S$\leftarrow \emptyset$.

For each $\delta = \pm 1$, $j = 1, \ldots, 7$,

Initialize Q as an empty queue of boxes.

$\mathbf{d} \leftarrow \delta \mathbf{e}_j$.

if (princ$_{\mathbf{d}}(B) =$ `NULL`), continue.

Q.insert(princ$_{\mathbf{d}}(B)$).

While (!Q.empty())

B_1 $\leftarrow$Q.pop().

For each child B_2 of B_1 in direction rev$_B(\mathbf{d})$,

if (B_2 is leaf) and ($B \xrightarrow{\mathbf{d}} B_2$)

S.insert(B_2).

else if ($B \xrightarrow{\mathbf{d}} B_2$)

Q.insert(B_2).

return S.

The process AllNeighbor(B) always halts since there are finite children in princ$_{\mathbf{d}}(B)$. It is correct by Lemma 6.2.7.

Chapter 7

Conclusion and Future Plan

This thesis has completed the SSS Axioms and proven the Fundamental Theorem of the SSS framework, thereby concluding the series of work initiated in [59]. Based on these Axioms, the thesis designed approximate footprints and subdivision schemes for the Delta robot $\triangle\mathcal{AOB}$ and demonstrated that the designed SSS planner for the Delta robot satisfies all five Axioms in the SSS framework. Moreover, this planner is the first explicit, certified, and practical path planner for an $SE(3)$ robot that also does not have the halting problem. It serves as proof that the SSS framework is a truly practical method.

We have made an exhaustive analysis of the resolution of the Delta robot, and implemented the methods in this thesis. We discuss the results in this chapter.

7.1 Resolution Constant of the Delta Robot

The theory for the Delta robot $\triangle\mathcal{AOB}$ can be fully deduced by the Fundamental Theorem 2.2.6. By Proposition 6.1.2, the subdivision constant $D_0 = 9\sqrt{3}$. By Proposition 3.3.1, the atlas constant $C_0 = 2$. By Theorem 4.1.3, the Lipschitz constant $L_0 = 1$. By Theorem 4.2.2, the effective constant $\sigma = \frac{3}{2}D_0$. Therefore, by Theorem 2.2.6, the resolution constant

$$K = \max\{L_0C_0D_0\sigma, 4D_0^2L_0C_0, 4\} = 8D_0^2 = 1944.$$

This number is quite huge, however we have a more accurate estimation. Our subdivision method only applies box subdivisions, and therefore, for the resolution lower bound, we can apply Lemma A.6.1. Moreover, Lemma 2.2.9 is given by a uniformly ε subdivision. Usually, our $\varepsilon < 0.125$. By Corollary 6.1.3, our ε-small boxes in the subdivision scheme applied to the Delta robot have maximum aspect ratio $\sqrt{6}$. Therefore, the atlas constant can be reduced to

$$K = \max\{L_0 C_0 D_0 \sigma, 2\} = \max\{3D_0^2, 2\} = 18.$$

We summarize the conclusion as Theorem 7.1.1.

Theorem 7.1.1. *The resolution constant for the Delta robot is $K = 18$.*

7.2 Performance Analysis

Let $r_0 = 1$ which is the length of the legs of $\triangle\mathcal{AOB}$. Ideally, to work out a valuable scene that requires the robot to rotate to avoid the obstacles, the clearance of the possible paths in the scene should be fewer than r_0. The Fundamental Theorem depicts that we will output a path when there is a path with clearance $K\varepsilon$ where $K = 36$. Hence for the scenarios requiring the robot to fully rotate, ε should be small enough such that $K\varepsilon < r_0 = 1$, or equivalently $\varepsilon < \frac{1}{18}$. Suppose that the range of the environment has width $\mathrm{w}(B_0^t)$, to get an ε small box, one need to split $m + (n+1)$ depths in the subdivision tree such that $2^m > 18\mathrm{w}(B_0^t)$ and $2^n > 18$ where m is the depths of T-split and $n+1$ is the depths of R-split. The solution gives $n = 5$ and $m = 6$ even when W is small like 3 which allows the Delta robot to translately move in the environment.

The subdivision tree for the Delta robot is generally an oct-tree, except for the first R-split. In the worst case, there are $4 \times 8^{m+n}$ boxes for a uniformly ε subdivision. Based on the estimation in the last paragraph, $m + n = 11$, where the amount of boxes raises up to $2^{35} \approx 3.4 \times 10^{10}$. This number of boxes makes exhaustive traversal infeasible.

A **heuristic** is a priority function that dictates priority of each `MIXED` leaf in the subdivision tree, which applies to the priority queue Q in the SSS framework. This priority plays a key role in accelerating the subdivision search to avoid the thorough traverse in the subdivision tree. We have attempted several heuristics. Some of the heuristics give considerable efficiency. See Figure 7.1.

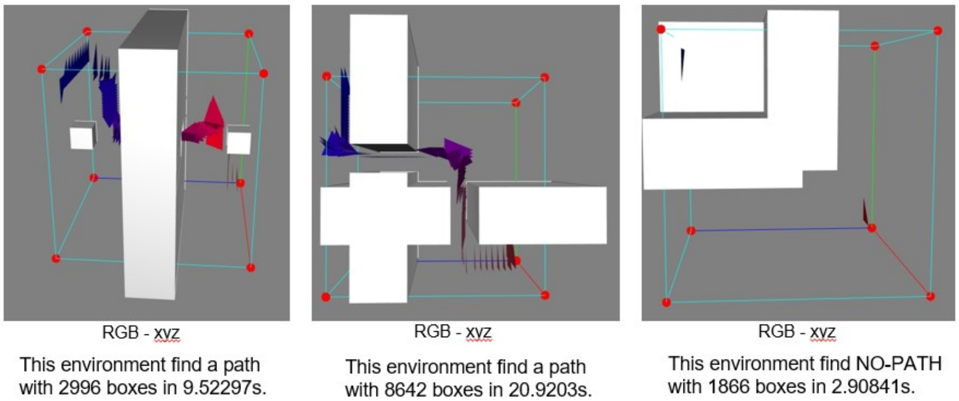


Figure 7.1: Performance of the algorithm (Demo).

See more data in Appendix F.

7.3 Future Work

The limitations of this research are primarily related to efficiency, which appears suboptimal. The heuristic function is critical to the planner's performance. Indeed, the search strategy is the main factor influencing the efficiency of an SSS planner, and a well-optimized strategy is still needed for the Delta robot. The near future work will focus on this problem.

Beyond the Delta robot, future work will extend the framework to accommodate arbitrary rigid-body robots. A challenge is to design appropriate approximate footprints. The approximate footprints for the Delta robot are the convex hull of the balls containing the exact footprints of all corners. A straightforward extension would involve designing approximate footprints for rigid-body robots again by the convex hull of the balls containing the exact footprints of all corners in the robot. A very primitive extension is the tetrahedron robot, which will be the first type that is researched. Subsequently, we will explore efficient methods to decompose arbitrary rigid body robots into tetrahedrons and then design a general approximate footprint strategy for those robots. An ultimate experimental target is the alpha puzzle [64].

Kinematic path planning is the immediate application of the SSS framework. We also look forward to extending the SSS framework into the kyno-dynamic path planning area. Unlike the configuration space in the kinematic path planning, the state space for kyno-dynamic path planning is not purely a Euclidean typed space (where each component is independent to another). There are extra relations between different components like between $\mathbf{x}$ and $\dot{\mathbf{x}}$. The appropriate definition for the footprint map is a challenge and it is also hard to design subdivisions in the state space. Future research will also combine with soft-exact methods for solving differential equations, which is a potential footprint map in the state space.

Appendix A

Configuration Space, Physical Space, and SSS framework

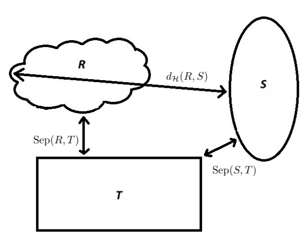
Configuration space and physical space are two fundamental elements in the definition of footprint map. They play key roles in the path planning problem. To guarantee stronger results, we need to fully understand the conditions that the two spaces are required to have.

A.1 Triangle Inequality

This section proves the triangle inequality between Separation and Hausdorff distance.

Lemma A.1.1 (Triangle Inequality)**.** *Let $\mathcal{Z}$ be a metric space. Then for any subsets $R, S, T \subseteq \mathcal{Z}$,*

$$|Sep(R,T) - Sep(S,T)| \leq d_{\mathcal{H}}(R,S).$$



Proof. Let $d_{\mathcal{Z}}$ be the metric of $\mathcal{Z}$. Then,

$$\begin{aligned}
\mathrm{Sep}(R,T) &= \inf_{r\in R} \mathrm{Sep}(\{r\},T) \\
&= \inf_{r\in R} \left(\inf_{t\in T} d_{\mathcal{Z}}(r,t) \right) \\
&\leq \inf_{r\in R} \left(\inf_{t\in T} \left(d_{\mathcal{Z}}(r,s) + d_{\mathcal{Z}}(s,t) \right) \right),\ \forall s \in S \\
&\leq \inf_{r\in R} \left(d(r,s) + \inf_{t\in T} d(s,t) \right),\ \forall s \in S \\
&= \inf_{r\in R} \left(d(r,s) + \mathrm{Sep}(\{s\},T) \right),\ \forall s \in S \\
&= \inf_{r\in R} d(r,s) + \mathrm{Sep}(\{s\},T),\ \forall s \in S \\
&= \mathrm{Sep}(R,\{s\}) + \mathrm{Sep}(\{s\},T),\ \forall s \in S
\end{aligned}$$

Taking the inferior of the right side, we get

$$\begin{aligned}
\mathrm{Sep}(R,T) &\leq \inf_{s\in S} (\mathrm{Sep}(R,\{s\}) + \mathrm{Sep}(\{s\},T)) \\
&\leq \sup_{s\in S} \mathrm{Sep}(R,\{s\}) + \inf_{s\in S} \mathrm{Sep}(\{s\},T) \\
&\leq d_{\mathcal{H}}(R,S) + \mathrm{Sep}(S,T)
\end{aligned}$$

And hence, $\mathrm{Sep}(R,T) - \mathrm{Sep}(S,T) \leq d_{\mathcal{H}}(R,S)$. Similarly, $\mathrm{Sep}(S,T) - \mathrm{Sep}(R,T) \leq d_{\mathcal{H}}(S,R) = d_{\mathcal{H}}(R,S)$. Therefore,

$$|\mathrm{Sep}(R,T) - \mathrm{Sep}(S,T)| \leq d_{\mathcal{H}}(R,S).$$

□

A.2 Manifold Structure in Configuration Space

In the SSS framework, the finding path process is to find a path π connecting α and β. It is achieved by finding a `FREE` channel covering the proposed path π. Specifically, when $\mathcal{X}$ is a manifold, since $\mathcal{Y}$ is an open subset of $\mathcal{X}$, it is still a manifold. This guarantees the existence of the channel if $\alpha, \beta \in \mathcal{Y}$ are in the same path-connected component, as ensured by the Poincaré's Duality. In this section, we discuss the role of Poincaré's Duality in the manifold Cspace.

The Poincaré's Duality guarantees a path in the dual manifold which is the channel aimed by the SSS planner. We here give a brief review of Poincaré's Duality [42]:

Poincaré's Duality is an important theorem in algebraic topology and it is described in standard textbooks. We use Munkres [42] as our main reference. Munkres [42] describes Poincaré's Duality in a **compact triangulated homology manifold**. Naively, it is a union of essentially disjoint compact simplices. It is analogous to the subdivision of the boxes, where a subdivision of a box is a union of essentially disjoint compact boxes. Given a subdivision tree rooted at B_0, the corners and edges of the subboxes of B_0 form a graph, denoted by $\mathrm{G}(B_0)$. The duality implies the existence of a path in the dual graph of the graph $\mathrm{G}(B_0)$. This path in the dual graph is the channel that an SSS planner aims to find. The formalized proof of this process can be easily and rigorously derived by a corollary of the Poincaré's Duality in [42]. We take it as our core lemma:

Lemma A.2.1 (Munkres 65.2)**.** *Let X be a compact triangulated homology n-manifold. If X is connected, then for any two n-simplices σ, σ' of X, there is a sequence*

$$\sigma = \sigma_0, \sigma_1, \ldots, \sigma_m = \sigma'$$

of n-simplices of X such that $\sigma_i \cap \sigma_{i+1}$ is an $n-1$ simplex of X for each i.

For simplicity, we define two simplices/boxes to be **adjacent** if they share a codimension 1 facet, as in Lemma A.2.1.

If $\mathcal{X}$ is a manifold, by the representation map, $B_0 = \mathrm{dom}(\mu) = \mu^{-1}(\mathcal{X})$ is also a manifold, and it is a union of boxes. Given a subdivision tree $\mathcal{T}(B_0)$, for any $\alpha, \beta \in B_0$ and a path $\pi \subseteq B_0$ such that $\pi(0) = \alpha$ and $\pi(1) = \beta$, we call the collection of leaves in $\mathcal{T}(B_0)$ that covers π the **cover** of π, denoted by $\texttt{Cover}(\pi)$, i.e.,

$$\texttt{Cover}(\pi) = \{B \in \mathcal{L}(\mathcal{T}(B_0)) : B \cap \pi \neq \emptyset\}.$$

The SSS framework aims to find a channel in $\texttt{Cover}(\pi)$, if it exists. The existence of the channel is based on an "observation", an "operation" and an "extension".

The first is an "observation", which gives Proposition A.2.2.

Proposition A.2.2 (Cover of path). *For any path $\pi \subseteq B_0$ and any finite subdivision of B_0, the union of collection $\texttt{Cover}(\pi)$ is a connected compact manifold.*

Proof. The space $\texttt{Cover}(\pi)$ as a subset of $\mathcal{L}(\mathcal{T}(B_0))$ is always a compact space. Let $\texttt{Cover}(\pi)^\circ$ be the interior of $\texttt{Cover}(\pi)$, which is a manifold as an open subset of B_0. Moreover, for each point $\gamma \in \pi$, γ is always an inner point, since if γ is not the inner point of some box, we can construct an open ball with radius less than the minimum width in $\mathcal{L}(\mathcal{T}(B_0))$, where the minimum exists since the subdivision is finite. The open ball will be contained in a union of boxes in $\texttt{Cover}(\pi)$. Hence $\pi \subseteq \texttt{Cover}(\pi)^\circ$ which implies $\texttt{Cover}(\pi)^\circ$ is path-connected. Hence $\texttt{Cover}(\pi)$ is a connected compact manifold. □

The second is an "operation", which results in Theorem A.2.3.

Theorem A.2.3. *The boxes in a subdivision tree of a manifold B_0 can be decomposed into triangulated homology manifold C_0 such that*

1. *Each box is decomposed into a connected triangulated manifold where each simplex σ in C_0 is uniquely contained in a box B in B_0;*

2. *If σ_i and σ_j are adjacent simplices in C_0, where σ_i is a child of B_i and σ_j is a child of B_j for some distinct boxes B_i and B_j in M, then B_i and B_j are adjacent.*

Proof. Suppose that we have a subdivision $B_0 = \cup_{i=1}^m B_i$. We use Lexicographic Triangulations [31] to decompose each B_i and result in $C_0 = \cup_{j=1}^n \sigma_j$. Since each box is a convex polytope, it is always a connected manifold and the triangulation makes it into a connected triangulated manifold. The triangulation takes place respectively in each box and hence each simplex σ is contained in a unique box B. For any adjacent simplices σ_i and σ_j in C_0, $\sigma_i \cap \sigma_j$ is a shared facet with codimension 1. Then, as a result,

$\dim(M) > \dim(B_i \cap B_j) \geq \dim(\sigma_i \cap \sigma_j) = \dim(M) - 1$ since $\sigma_i \cap \sigma_j \subseteq B_i \cap B_j$ and B_i is distinct with B_j. Hence $\dim(B_i \cap B_j) = \dim(M) - 1$ and B_i and B_j are adjacent. □

The third extension gives Corollary A.2.4.

Corollary A.2.4. *Let B_0 be a compact box (homology) d-manifold. If B_0 is connected, then for any two d-boxes B, B' of B_0, there is a sequence*

$$B = B_1, B_2, \ldots, B_n = B'$$

of d-boxes of B_0 such that $B_i \cap B_{i+1}$ is a $(d-1)$-facet of B_0 for each i.

Proof. Let C_0 be the compact triangulated d-manifold described in theorem A.2.3. By Lemma A.2.1, there is a sequence

$$\sigma = \sigma_1, \sigma_2, \ldots, \sigma_m = \sigma'$$

of d-simplices of C_0 such that $\sigma_j \cap \sigma_{j+1}$ is a $(d-1)$-simplex of C_0 for each j.

We do two operations on the sequence $\sigma_1, \sigma_2, \ldots, \sigma_m$:

1. For each σ_j in the sequence, let B_i be the unique box that contains it, substitute σ_i by B_i in the sequence;

2. For each loop shaping $B_i = B_{i_0}, B_{i_1}, B_{i_2}, \ldots, B_{i_l} = B_i$, substitute by B_i.

Note that step 2. also removes any consecutively repeating B_i.

After the two operations, let the sequence $\sigma_1, \sigma_2, \ldots, \sigma_m$ become $B_1, B_2, \ldots, B_n$. Since B_i and B_{i+1} are generated from adjacent σ_j and σ_{j+1} by operations for each i, by Theorem A.2.3, B_i and B_{i+1} are also adjacent. Therefore, we have constructed the sequence as required in the corollary. □

As a summary, the cover of any path π in any finite subdivision tree can be triangulated into a triangulated homology manifold, where we can always construct a channel connecting any pair of simplices, from which we can construct a corresponding channel for the SSS planner.

A.3 Structure of Boxes

The subdivision takes place on tiles in $\square\mathcal{W}$. In general, a **tile** B is the convex hull of a set of points $\mathbf{w}_1, \ldots, \mathbf{w}_n \in \mathcal{W}$, denoted by $B = \mathrm{Chull}\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$. The **relative center** of the tile B is its center of gravity, denoted by $\mathrm{m}(B) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{w}_i$. Given any $\sigma > 0$, we define a **σ-homothet** of B as

$$\sigma B = \mathrm{Chull}_{i=1}^{n}\{\mathrm{m}(B) + \sigma(\mathbf{w}_i - \mathrm{m}(B))\},$$

see Figure A.1 for a sketch map.

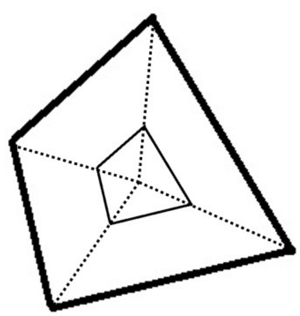


Figure A.1: A homothet of a tile.

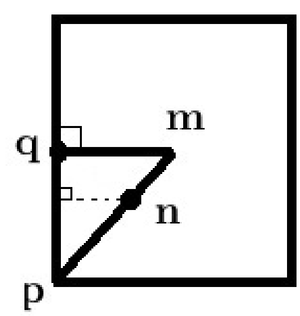


Figure A.2: Aspect ratio in Lemma A.3.2

In practice, we usually do not use arbitrary polytopes in subdivisions. The choice of polytopes depends on the subdivision scheme. Let $\square\mathcal{W}$ denote a set of polytopes. For instance, $\square\mathcal{W}$ may be the set of all simplices, or the set of all boxes. Recall that in Chapter 1, we have restricted our tiles into such boxes and defined the **width** of a box $B = \prod_{i=1}^{d} I_i$ to be $\mathrm{w}(B) = \min_{i=1}^{d}|I_i|$ and the **length** of B is $\ell(B) = \sup_{\mathbf{x},\mathbf{y}\in B} d_{\mathcal{W}}(\mathbf{x},\mathbf{y})$. The **aspect ratio** is $\alpha(B) = \ell(B)/\mathrm{w}(B)$. By definition, $\forall \mathbf{x}, \mathbf{y} \in B$, $d_{\mathcal{W}}(\mathbf{x},\mathbf{y}) \le \ell(B)$. The relation between a box B and its homothet σB includes:

Proposition A.3.1.

$$w(\sigma B) = \sigma w(B),\ \ \ell(\sigma B) = \sigma\ell(B),\ \ \alpha(\sigma B) = \alpha(B).$$

Another method of estimating the clearance of a path, we introduce the lemma A.3.2 of aspect ratio:

Lemma A.3.2. *For any box $B \in \square\mathcal{W}$ and any point $\mathbf{p} \in \partial B$, the sine of angle between the line segment connecting $\mathbf{p}$ with $\mathbf{m} = m(B)$ and any facet containing $\mathbf{p}$ is lower bounded by $1/\alpha(B)$, i.e., if F is the facet of B containing $\mathbf{p}$,*

$$\frac{Sep(\{\mathbf{m}\}, F)}{d_{\mathcal{W}}(\mathbf{m}, \mathbf{p})} \geq \frac{1}{\alpha(B)}.$$

Proof. Since $\mathbf{m}$ is the relative center, as a result, $\mathrm{Sep}(\{\mathbf{m}\}, F) \geq \frac{1}{2}\mathrm{w}(B)$ and $d_{\mathcal{W}}(\mathbf{m}, \mathbf{p}) \leq \frac{1}{2}\ell(B)$. Hence

$$\frac{\mathrm{Sep}(\{\mathbf{m}\}, F)}{d_{\mathcal{W}}(\mathbf{m}, \mathbf{p})} \geq \frac{\mathrm{w}(B)}{\ell(B)} = \frac{1}{\alpha(B)}.$$

□

The similarity of triangles implies corollary A.3.3.

Corollary A.3.3. *For any box $B \in \square\mathcal{W}$ and any point $\mathbf{p} \in \partial B$, for any $\mathbf{n}$ in the line segment $\overline{\mathbf{pm}}$,*

$$\frac{Sep(\{\mathbf{n}\}, \partial B)}{d_{\mathcal{W}}(\mathbf{n}, \mathbf{p})} \geq \frac{1}{\alpha(B)}.$$

Remark A.3.4. The Lemma A.3.2 and Corollary A.3.3 can be extended to any tile if $\mathrm{w}(B)$ and $\ell(B)$ are defined properly. The SSS framework may not be restricted to subdivision of purely boxes. The Fundamental Theorem for general tile subdivisions is still correct. But this will be future work.

A.4 Linear Physical Space

The physical space $\mathcal{Z}$ is generally a metric space. But in practice, it is a subset of the universe which is locally Euclidean. The Euclidean spaces are linear spaces where we can use vector operations to design approximate footprints. In the SSS framework, $\mathcal{Z} = \mathbb{R}^k$ for some positive integer k, which is a linear space. The linear space have Minkowski sum operation.

Given a linear space $\mathcal{Z}$, the **Minkowski sum** on $\mathcal{I}, \mathcal{K} \subseteq \mathcal{Z}$ is defined as

$$\mathcal{I} \oplus \mathcal{K} = \{\iota + \kappa : \iota \in \mathcal{I}, \kappa \in \mathcal{K}\}.$$

The key usage of Minkowski sum is to estimate the clearance of configurations. A set $\mathcal{I}$ is called a **fat** $\mathcal{K}$, if $\mathcal{I} = \mathcal{K} \oplus \mathrm{Ball}(s)$ for some $s > 0$. Usually, we only require $\mathrm{Cl}(\gamma) > 0$ for a configuration γ to be

FREE. But we can simplify the computation of checking $\mathrm{Cl}(\gamma) > 0$ when $\mathrm{Fp}(\gamma)$ is a fat polyhedral set. The process called **parametric reduction** is based on Lemma A.4.1:

Lemma A.4.1. *Let $\mathcal{E}, \Omega \subseteq \mathbb{R}^k = \mathcal{Z}$ be closed sets. Then $Sep(\mathcal{E} \oplus Ball(s), \Omega) > 0$ if and only if $Sep(\mathcal{E}, \Omega) > s$.*

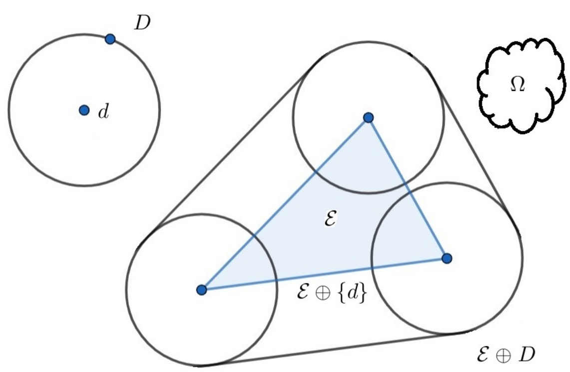
Proof.

$$\begin{aligned}
\mathrm{Sep}(\mathcal{E} \oplus \mathrm{Ball}(s), \Omega) > 0 &\Leftrightarrow \forall \mathbf{e} \in \mathcal{E}, \forall \mathbf{f} \in \Omega, \mathrm{Sep}(\{\mathbf{e}\} \oplus \mathrm{Ball}(s), \{\mathbf{f}\}) > 0 \\
&\Leftrightarrow \forall \mathbf{e} \in \mathcal{E}, \forall \mathbf{f} \in \Omega, d_{\mathcal{H}}(\{\mathbf{e}\}, \{\mathbf{f}\}) = d_{\mathcal{Z}}(\mathbf{e}, \mathbf{f}) > s \\
&\Leftrightarrow \mathrm{Sep}(\mathcal{E}, \Omega) > s.
\end{aligned}$$

□

The lemma A.4.1 estimates the separation $\mathrm{Sep}(\mathcal{E}, \Omega)$ based on knowing $\mathrm{Sep}(\mathcal{E} \oplus D, \Omega) > 0$ when D is a ball. We can extend this result to the case where D is not a ball, but any closed set. It results in Lemma A.4.2.

Lemma A.4.2 (Separation of Sum)**.** *Suppose that closed sets $D, \mathcal{E}, \Omega \subseteq \mathbb{R}^k = \mathcal{Z}$. If $Sep(\mathcal{E} \oplus D, \Omega) > 0$, then for any $\mathbf{d} \in D$, $Sep(\mathcal{E} \oplus \{\mathbf{d}\}, \Omega) \geq Sep(\{\mathbf{d}\}, \partial D)$.*



Proof. We denote the complement of a set S in $\mathbb{R}^k$ by $\mathbb{R}^k/S$. $\mathrm{Sep}(\mathcal{E} \oplus D, \Omega) > 0$ implies that $\Omega \subseteq \mathbb{R}^k/(\mathcal{E} \oplus D)$. Since $\mathcal{E} \oplus \{\mathbf{d}\} \subseteq \mathcal{E} \oplus D$, we have

$$\mathrm{Sep}(\mathcal{E} \oplus \{\mathbf{d}\}, \Omega) \geq \mathrm{Sep}(\mathcal{E} \oplus \{\mathbf{d}\}, \mathbb{R}^k/(\mathcal{E} \oplus D))$$

$$
\begin{aligned}
&= \mathrm{Sep}(\mathcal{E} \oplus \{\mathbf{d}\}, \partial(\mathbb{R}^k/(\mathcal{E} \oplus D))) \\
&= \mathrm{Sep}(\mathcal{E} \oplus \{\mathbf{d}\}, \partial(\mathcal{E} \oplus D)) \\
&= \inf_{(\mathbf{e}'+\mathbf{d}') \in \partial(\mathcal{E} \oplus D)} \inf_{\mathbf{e} \in \mathcal{E}} d_{\mathcal{Z}}(\mathbf{e}+\mathbf{d}, \mathbf{e}'+\mathbf{d}') \\
&\geq \inf_{\mathbf{d}' \in \partial D} \inf_{\mathbf{e} \in \mathcal{E}} \inf_{\mathbf{e}' \in \partial \mathcal{E}} d_{\mathcal{Z}}(\mathbf{e}+\mathbf{d}, \mathbf{e}'+\mathbf{d}') \\
&\geq \inf_{\mathbf{d}' \in \partial D} \inf_{\mathbf{e} \in \mathcal{E}} d_{\mathcal{Z}}(\mathbf{e}+\mathbf{d}, \mathbf{e}+\mathbf{d}') \\
&= \inf_{\mathbf{d}' \in \partial D} d_{\mathcal{Z}}(\mathbf{d}, \mathbf{d}') \\
&= \mathrm{Sep}(\{\mathbf{d}\}, \partial D).
\end{aligned}
$$

□

Minkowski Sum of convex sets can be easily computed as Lemma A.4.3.

Lemma A.4.3. *Let $\mathcal{K}$ be a convex set, $\mathcal{I}_i \subseteq \mathbb{R}^k$ for $i = 1, \ldots,$. Then*

$$\mathcal{K} \oplus \mathit{Chull}\{\mathcal{I}_i\} = \mathit{Chull}\{\mathcal{K} \oplus \mathcal{I}_i\}.$$

Proof.

$$
\begin{aligned}
\mathcal{K} \oplus \mathrm{Chull}\{\mathcal{I}_i\} &= \left\{ \kappa + \sum_i a_i \iota_i : \kappa \in \mathcal{K}, \iota_i \in \mathcal{I}_i, a_i \in [0,1], \sum_i a_i = 1 \right\} \\
&= \left\{ \sum_i a_i(\kappa + \iota_i) : \kappa \in \mathcal{K}, \iota_i \in \mathcal{I}_i, a_i \in [0,1], \sum_i a_i = 1 \right\} \\
&= \mathrm{Chull}\{\mathcal{K} \oplus \mathcal{I}_i\}.
\end{aligned}
$$

□

A.5 Effective Conditions

A σ-effective soft predicate gives sufficient and necessary conditions for declaring a positive clearance as in Lemma A.5.1 and Lemma A.5.2.

Lemma A.5.1 (Effective Sufficient Condition)**.** *If the soft predicate $\widetilde{C}$ is σ-effective, then a box $B \in \square\mathcal{W}$ is* `FREE` *in SSS process if for each point $b \in \sigma B$, $Cl(b) > 0$.*

Proof. We see these relations:

$$\begin{aligned} \forall b \in \sigma B,\ Cl(b) > 0 &\Rightarrow C(\sigma B) = \texttt{FREE} \\ &\Rightarrow \widetilde{C}(B) = \texttt{FREE} \\ &\Rightarrow B \text{ is } \texttt{FREE} \text{ in SSS.} \end{aligned}$$

□

Lemma A.5.2 (Effective Necessary Condition)**.** *For soft predicate $\widetilde{C}$, a box $B \in \square\mathcal{W}$ is* `FREE` *in SSS process only if for each point $\mathbf{b} \in B$, $Cl(\mathbf{b}) > 0$.*

Proof. We see these relations:

$$\begin{aligned} B \text{ is } \texttt{FREE} \text{ in SSS} &\Rightarrow \widetilde{C}(B) = \texttt{FREE} \\ &\Rightarrow C(B) = \texttt{FREE} \\ &\Rightarrow \forall \mathbf{b} \in B,\ \mathrm{Cl}(\mathbf{b}) > 0. \end{aligned}$$

□

A.6 Translational Lemma for More Accurate Resolution

The translational lemma in Lemma 2.2.10 can be easily extended to general tile subdivisions. In fact, all lemmas in the proof are exactly applicable to general tiles. For the special box space subdivisions, we have a better translational lemma with a much more accurate resolution. In fact, the resolution constant for the (N) part can be reduced to exactly 2. Let us see Lemma A.6.1.

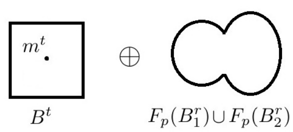
Lemma A.6.1. *If an SSS planner applies only box subdivisions that satisfies **(A0)** and **(A4)**, then the there is a path in $\mu^{-1}(C_{free})$ that has essential clearance $\frac{\varepsilon}{K}$ for $K = 2$ if the SSS planner outputs a path.*

Proof. Suppose that $\mathcal{T}(B_0)$ is the subdivision tree when the SSS planner outputs a path. We continue splitting $\mathcal{T}(B_0)$ until it becomes ε-uniform. Then we can still find a channel in this ε-uniform subdivision tree. We pick one channel P' and construct the canonical path $\overline{P}'$ from P'. We show that $\overline{P}'$ has essential clearance $\frac{\varepsilon}{2}$.

Let $B_1 = B_1^t \times B_1^r$ and $B_2 = B_2^t \times B_2^r$ be the two adjacent boxes in P', where $B_1^t, B_2^t \subseteq \mathcal{W}^t = \mathcal{Z}$. Their centers are $\mathbf{m}_1 = \mathbf{m}_1^t \times \mathbf{m}_1^r$ and $\mathbf{m}_2 = \mathbf{m}_2^t \times \mathbf{m}_2^r$ respectively. The line segment connecting $\mathbf{m}_1$ and $\mathbf{m}_2$ is $\overline{\mathbf{m}_1\mathbf{m}_2}$. Since $\dim(B_1 \cap B_2) = d - 1$, either $B_1^t = B_2^t$ or $B_1^r = B_2^r$. We discuss these two cases respectively.

- When $B_1^t = B_2^t = B^t$, $\mathbf{m}_1^t = \mathbf{m}_2^t = \mathbf{m}^t$. For any $\mathbf{b} \in \overline{\mathbf{m}_1\mathbf{m}_2}$, we write $\mathbf{b} = \mathbf{m}^t \times \mathbf{b}^r$. Since, B_1 and B_2 are `FREE` boxes,

$$
\begin{aligned}
\widetilde{C}(B_1) = \widetilde{C}(B_2) = \texttt{FREE} &\Rightarrow \mathrm{Fp}(B_1) \cap \Omega = \emptyset \text{ and } \mathrm{Fp}(B_2) \cap \Omega = \emptyset \mathbf{(A0)} \\
&\Rightarrow (\mathrm{Fp}(B_1) \cup \mathrm{Fp}(B_2)) \cap \Omega = \emptyset \\
&\Rightarrow \big((B^t \oplus \mathrm{Fp}(B_1^r)) \cup (B^t \oplus \mathrm{Fp}(B_2^r)) \big) \cap \Omega = \emptyset \mathbf{(A4)} \\
&\Rightarrow \big(B^t \oplus (\mathrm{Fp}(B_1^r) \cup \mathrm{Fp}(B_2^r)) \big) \cap \Omega = \emptyset
\end{aligned}
$$



As $\mathbf{m}^t \in B^t$, $\mathrm{Fp}(\overline{\mathbf{m}_1^r\mathbf{m}_2^r}) \subseteq (\mathrm{Fp}(B_1^r) \cup \mathrm{Fp}(B_2^r))$. By Lemma A.4.2,

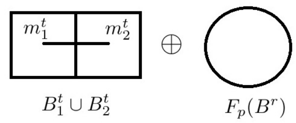
$$
\mathrm{Cl}(\overline{\mathbf{m}_1\mathbf{m}_2}) = \mathrm{Sep}(\mathbf{m}^t \oplus \mathrm{Fp}(\overline{\mathbf{m}_1^r\mathbf{m}_2^r}), \Omega) \geq \mathrm{Sep}(\mathbf{m}^t \oplus (\mathrm{Fp}(B_1^r) \cup \mathrm{Fp}(B_2^r)), \Omega) \geq \mathrm{Sep}(\{\mathbf{m}^t\}, \partial B^t) \geq \frac{1}{2}\varepsilon.
$$

- When $B_1^r = B_2^r = B^r$, $\mathbf{m}_1^r = \mathbf{m}_2^r = \mathbf{m}^r$. For any $\mathbf{b} \in \overline{\mathbf{m}_1\mathbf{m}_2}$, we write $\mathbf{b} = \mathbf{b}^t \times \mathbf{m}^r$. Since, B_1 and B_2 are `FREE` boxes,

$$
\begin{aligned}
\widetilde{C}(B_1) = \widetilde{C}(B_2) = \texttt{FREE} &\Rightarrow \mathrm{Fp}(B_1) \cap \Omega = \emptyset \text{ and } \mathrm{Fp}(B_2) \cap \Omega = \emptyset \mathbf{(A0)} \\
&\Rightarrow (\mathrm{Fp}(B_1) \cup \mathrm{Fp}(B_2)) \cap \Omega = \emptyset
\end{aligned}
$$

$$\Rightarrow \big(\big(B_1^t \oplus \mathrm{Fp}(B^r)\big) \cup \big(B_2^t \oplus \mathrm{Fp}(B^r)\big)\big) \cap \Omega = \emptyset \textbf{(A4)}$$

$$\Rightarrow \big(\big(B_1^t \cup B_2^t\big) \oplus \mathrm{Fp}(B^r)\big) \cap \Omega = \emptyset.$$



As $\mathbf{b} \in B_1^t \cup B_2^t$, by Lemma A.4.2,

$$\mathrm{Cl}(\overline{\mathbf{m}_1\mathbf{m}_2}) = \inf_{\mathbf{b}\in\overline{\mathbf{m}_1\mathbf{m}_2}} \mathrm{Sep}(\mathbf{b}^t \oplus \mathrm{Fp}(B^r), \Omega) \geq \inf_{\mathbf{b}\in\overline{\mathbf{m}_1\mathbf{m}_2}} \mathrm{Sep}(\{\mathbf{b}^t\}, \partial(B_1^t \cup B_2^t)) = \mathrm{Sep}(\overline{\mathbf{m}_1^t\mathbf{m}_2^t}, \partial(B_1^t \cup B_2^t)) \geq \frac{1}{2}\varepsilon.$$

Both of the two cases can deduce that $\mathrm{Cl}(\overline{\mathbf{m}_1\mathbf{m}_2}) \geq \frac{1}{2}\varepsilon$ for any pair of adjacent boxes B_1, B_2 in P'. Therefore, the path $\overline{P}'$ has essential clearance $\frac{1}{2}\varepsilon$. $\square$

Appendix B

Differential Geometry and Distortion Bound Problems

B.1 An Error Related to Rod and Ring Robots

In this section, we give the intuition of the error in [23].

Example B.1.1. In [23], the footprint map of a rod robot ℓ is

$$\begin{aligned} \mathrm{Fp}_\ell : \mathbb{R}^3 \times S^2 &\to \mathcal{C}(\mathbb{R}^3) \\ (\mathbf{x}, \mathbf{v}) &\mapsto \{\mathbf{x} + t\mathbf{v} : t \in [0,1]\}. \end{aligned}$$

The rotational space S^2 is represented by

$$\begin{aligned} \mu^r : \partial[-1,1]^3 &\to S^2 \\ (x,y,z) &\mapsto (a,b,c) = \frac{1}{\sqrt{x^2+y^2+z^2}}(x,y,z). \end{aligned}$$

The paper states that $\sqrt{3}$ is a bound for the distortion of μ^r. Let $C = [-1,1]^3$ and B is the ball whose boundary is S^2. In the Appendix B.1. of [23], it states that $B \subseteq C \subseteq \sqrt{3}B$ and according to this to deduce that, for any geodesics $\alpha \subseteq \partial B = S^2$, $\alpha' = \sqrt{3}\alpha \subseteq \partial\sqrt{3}B$ and $\widehat{\alpha} \subseteq \partial C$ such that $\mu^r(\widehat{\alpha}) = \alpha$, $|\alpha| \leq |\widehat{\alpha}| \leq |\alpha'|$, which results in $1 \leq \frac{|\widehat{\alpha}|}{|\alpha|} \leq \sqrt{3}$ and $C_0 = \sqrt{3}$. However, the statement $|\widehat{\alpha}| \leq |\alpha'|$ is false. The proof can be given by the primary geometry.

Consider a plane crossing a diagonal of C that intersects B, C and $\sqrt{3}B$ simultaneously, see Figure B.1 for a sketch of the cut and Figure B.2 for the shapes on that cutting plane.

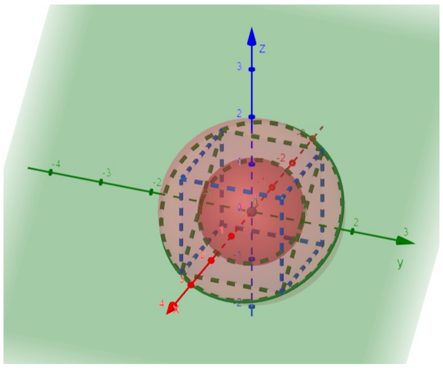


Figure B.1: A plane cutting B, C and $\sqrt{B}$ through a diagonal if C.

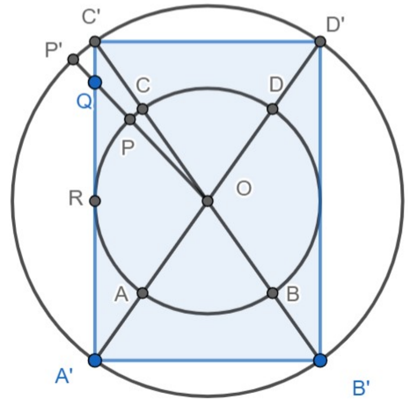


Figure B.2: The 2D shapes on the cut plane.

Let us estimate the distortion between P and C referring to the Figure B.2 when they are very close to each other. In the figure, the smaller circle with A, B, C and D represents the image of $B = S^2$ cut by the plane, and the larger circle with A', B', C' and D' represents the image of $\sqrt{3}B$ cut by the plane. The rectangle $A'B'D'C'$ represents the image of ∂C cut by the plane. The rotational component μ^r maps $\mu^r(C') = C$ and $\mu^r(Q) = P$. The distortion between P and C is approximately $\frac{\overline{PC}}{\overline{QC'}}$. When P and C are close enough, $\overline{P'P}$ is approximately parallel to $\overline{C'C}$ and $\overline{P'P}$ is approximately perpendicular to $\overline{P'C'}$. Then

$$\frac{\overline{PC}}{\overline{QC'}} = \frac{1}{\sqrt{3}}\frac{\overline{P'C'}}{\overline{QC'}} \approx \frac{1}{\sqrt{3}}\sin\angle P'Q'C' \approx \frac{1}{\sqrt{3}}\sin\angle QC'C = \frac{1}{\sqrt{3}}\frac{\overline{A'B'}}{\overline{B'C'}} = \frac{1}{3}.$$

The closer P is to C, the closer the distortion is $\frac{1}{3}$. Thus the distortion of μ^r is actually bounded by 3 where the atlas constant C_0 is at least 3 instead of $\sqrt{3}$.

B.2 Review of Differentiable Manifold

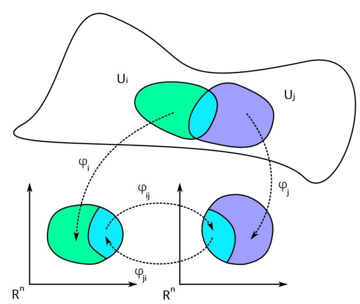
We restrict our Cspace to Riemannian manifolds which may be involved by most research in motion planning. The differential structures and Riemannian metrics are the keys to compute distortion bounds systematically.

A differentiable manifold is a topological manifold with a globally defined differential structure. We cite its description from a standard mathematics textbook [57].

Definition B.2.1. Let M be an n-dimensional manifold. M is a C^k **differentiable manifold**, if there is a countable open cover $U_i, i \in \mathbb{N}$ on M, which means $\cup_{i \in \mathbb{N}} U_i = M$, such that they satisfy the following properties:

1. For each U_i, there is a homeomorphic map $\phi_i : U_i \to \mathbb{R}^n$, each (U_i, ϕ_i) is called a **chart** (or local chart);

2. For any two charts (U_i, ϕ_i) and (U_j, ϕ_j), the **transition map** $\phi_{ij} = \phi_j \circ \phi_i^{-1}$ is a C^k map on $\phi_j(U_i \cap U_j)$.



Assume that both $\mathcal{W}$ and $\mathcal{X}$ to differentiable manifolds. When $\mathrm{dom}(\mu) = B_0 = \cup_{i=1}^{n} B_i$ and B_i are disjoint closed boxes, the differential structure on $\mathcal{W}$ requires an atlas of charts on each box B_i. The boxes B_i are closed, but charts are defined on open sets in the Definition B.2.1. To make it compatible, we extends the charts on B_i to an open neighborhood U_i of B_i, such that B_i is the unique box among $B_1, \ldots, B_n$ that makes $B_i \subseteq U_i$ for each i. As an example, we consider the representation of S^2 by

$$\mu : \partial[-1, 1]^3 \to S^2$$

$$\mathbf{x} \mapsto \frac{\mathbf{x}}{|\mathbf{x}|}$$

The dom(μ) is $B_0 = \widehat{S}^2 = \cup_{i=1}^6 B_i$ where

$$B_1 = \{-1\} \times [-1,1] \times [-1,1]$$

$$B_2 = \{1\} \times [-1,1] \times [-1,1]$$

$$B_3 = [-1,1] \times \{-1\} \times [-1,1]$$

$$B_4 = [-1,1] \times \{1\} \times [-1,1]$$

$$B_5 = [-1,1] \times [-1,1] \times \{-1\}$$

$$B_6 = [-1,1] \times [-1,1] \times \{1\}.$$

The chart defined on each B_i is a projection map by removing the constant component. See Figure B.3 and Figure B.4 for a reference which is the scene viewed from $-x$ direction. The constructions of such charts is given in Example B.2.2. The transition maps between different charts are differentiable which defines a differential structure on B_0.

Example B.2.2. We pick two adjacent faces as an example to show that their local charts can be extended to an open neighborhood where the transition map between the intersections are differentiable.

Let (U_1, ϕ_1) be the chart containing B_1, which is

$$\begin{aligned} U_1 = B_1 &\cup [-1,-0.8) \times \{-1\} \times [-1,1] \\ &\cup [-1,-0.8) \times \{1\} \times [-1,1] \\ &\cup [-1,-0.8) \times [-1,1] \times \{-1\} \\ &\cup [-1,-0.8) \times [-1,1] \times \{1\} \end{aligned}$$

such that

$$\phi_1 : U_1 \to \mathbb{R}^2$$

$$(-1, y, z) \mapsto (-y, z) \qquad \text{for } (-1, y, z) \in B_1$$

$$(x, -1, z) \mapsto (x+2)(1, z) \qquad \text{for } (x, -1, z) \in B_3$$

$$(x, 1, z) \mapsto (x+2)(-1, z) \qquad \text{for } (x, 1, z) \in B_4$$

$$(x, y, -1) \mapsto (x+2)(-y, -1) \qquad \text{for } (x, y, -1) \in B_5$$

$$(x, y, 1) \mapsto (x+2)(-y, 1) \qquad \text{for } (x, y, 1) \in B_6$$

Let (U_6, ϕ_6) be the chart containing B_6, which is

$$\begin{aligned} U_6 = B_6 &\cup \{-1\} \times [-1, 1] \times (0.8, 1] \\ &\cup \{1\} \times [-1, 1] \times (0.8, 1] \\ &\cup [-1, 1] \times \{-1\} \times (0.8, 1] \\ &\cup [-1, 1] \times \{1\} \times (0.8, 1] \end{aligned}$$

such that

$$\phi_6 : U_6 \to \mathbb{R}^2$$

$$(x, y, 1) \mapsto (-y, x) \qquad \text{for } (x, y, 1) \in B_6$$

$$(-1, y, z) \mapsto (z-2)(y, 1) \qquad \text{for } (-1, y, z) \in B_1$$

$$(1, y, z) \mapsto (z-2)(y, -1) \qquad \text{for } (1, y, z) \in B_2$$

$$(x, -1, z) \mapsto (z-2)(1, -x) \qquad \text{for } (x, -1, z) \in B_3$$

$$(x, 1, z) \mapsto (z-2)(-1, -x) \qquad \text{for } (x, 1, z) \in B_4$$

The transition maps between (U_1, ϕ_1) and (U_6, ϕ_6) is $\phi_{16} = \phi_6 \circ \phi_1^{-1}$, i.e.,

$$\phi_{16} : \mathbb{R}^2 \to \mathbb{R}^2$$

$$(x, y) \mapsto (y-2)(-x, 1) \qquad \text{for } (x, y) \in \phi_1(B_1 \cap U_6)$$

$$(x,y) \mapsto (x/y, y-2) \qquad \text{for } (x,y) \in \phi_1(U_1 \cap B_6)$$

$$(x,y) \mapsto (y/x-2)(1, -(x-2)) \qquad \text{for } (x,y) \in \phi_1(U_1 \cap U_6 \cap B_3)$$

$$(x,y) \mapsto -(y/x+2)(1, x+2) \qquad \text{for } (x,y) \in \phi_1(U_1 \cap U_6 \cap B_4)$$

The unique possible singular point for ϕ_{16} is $(x,y) = (0,0)$, which is not in $\phi_1(U_6)$. Hence ϕ_{16} is differentiable over $U_1 \cap U_6$.

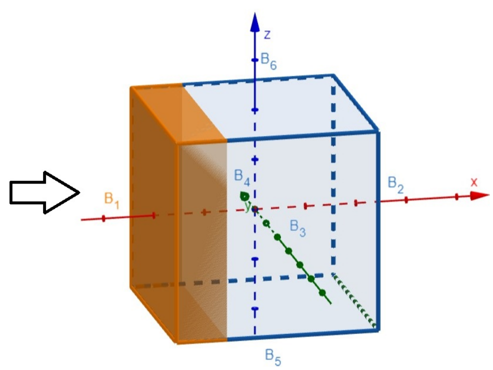


Figure B.3: Orange area is U_1.

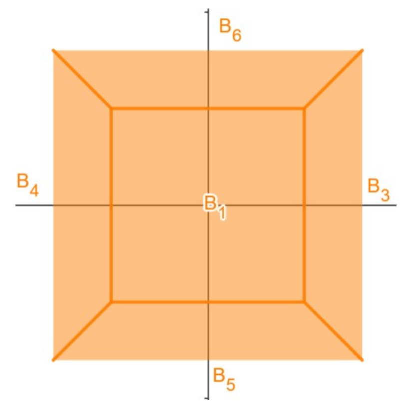


Figure B.4: Image of U_1 in $\mathbb{R}^2$

The differential structure on $\widehat{SO}(3) = C_w \cup C_x \cup C_y \cup C_z$ are similar.

Given differential structures on $\mathcal{W}$ and $\mathcal{X}$, the representation μ induces a push forward between the two differential structures. This push forward is defined by the correspondence between tangent spaces. Recall that a tangent space is a collection of tangent vectors. Given a differentiable manifold M, let $C^k(M)$ be the set of all k-differentiable functions on M. A **tangent vector** at a point $p \in M$ is a linear function v_p from $C^k(M) \to \mathbb{R}$, such that [39] for any $f, g \in C^k(M)$,

$$v_p(fg) = f(p)v_p(g) + g(p)v_p(f).$$

The set of all tangent vectors forms a linear space by defining $(av_p + bw_p)(f) = av_p(f) + bw_p(f)$ for any $f \in C^k(M)$ and any two tangent vectors v_p and w_p, which is the **tangent space** of M at point p, denoted by T_pM. For example, given a smooth curve $\pi : [0,1] \to M$, and $s \in [0,1]$, the map v_p for point

$p = \pi(s)$ defined by

$$v_p(f) = \frac{\mathrm{d}(f(\pi(t))}{\mathrm{dt}}\Big|_{t=s}$$

is a tangent vector of M at point p. This tangent vector is called the **derivative** of π at point p, denoted by $\pi'(s)$. The set of all curves π_i passing the point p identifying in each chart (U, ϕ) by $\pi_i \sim \pi_j$ if $\frac{\mathrm{d}\phi(\pi_i(t))}{\mathrm{dt}} = \frac{\mathrm{d}\phi(\pi_j(t))}{\mathrm{dt}}$ is the tangent space at p. For $\mathbf{b} \in \mathcal{W}$ and $\mu(\mathbf{b}) = \gamma \in \mathcal{X}$, the **push forward** of representation μ is a linear map $\mu_* : T_{\mathbf{b}}\mathcal{W} \to T_\gamma\mathcal{X}$ [39], such that for any $v_{\mathbf{b}} \in T_{\mathbf{b}}\mathcal{W}$ and for any $f \in C^d(\mathcal{X})$,

$$\mu_*(v_{\mathbf{b}})(f) = v_{\mathbf{b}}(f \circ \mu).$$

B.3 Riemannian Manifold and Distortion

A **Riemannian manifold** (M, g_M) is a differentiable manifold with a continuous bilinear function $g_M\langle\cdot,\cdot\rangle$ which called a **Riemannian metric** [9]. We restrict $\mathcal{W}$ and $\mathcal{X}$ to Riemannian manifolds. The representation μ induces a **pull back** μ^* that makes each Riemannian metric $g_{\mathcal{X}}$ on $\mathcal{X}$ into a Riemannian metric on $\mathcal{W}$ by

$$\mu^*(g_{\mathcal{X}})\langle v_{\mathbf{b}}, w_{\mathbf{b}}\rangle = g_{\mathcal{X}}\langle\mu_*(v_{\mathbf{b}}), \mu_*(w_{\mathbf{b}})\rangle.$$

Given a Riemannian manifold (M, g_M), the **length** of a tangent vector $v_p \in T_pM$ is

$$|v_p| = \sqrt{g_M\langle v_p, v_p\rangle}.$$

The length of a curve $\pi : [0,1] \to M$ is defined as

$$\ell(\pi) = \int_0^1 |\pi'(t)|\mathrm{dt} = \int_0^1 \sqrt{g_M\langle\pi'(t), \pi'(t)\rangle}\mathrm{dt}.$$

Given $p, q \in M$, the distance between p and q is defined as the length of the shortest curve connecting p and q, i.e.,

$$d_M(p,q) = \inf_{\pi:[0,1]\to M, \pi(0)=p, \pi(1)=q} \ell(\pi). \tag{B.1}$$

The metric on M is induced by this distance function d_M. When the manifolds are $\mathcal{W}$ and $\mathcal{X}$, the metrics $d_{\mathcal{W}}$ and $d_{\mathcal{X}}$ are also defined by (Equation B.1). Therefore, the distortion between $\mathbf{p}, \mathbf{q} \in d_{\mathcal{W}}$ where $\mu(\mathbf{p}) = \gamma$, $\mu(\mathbf{q}) = \zeta$ is

$$\frac{d_{\mathcal{X}}(\gamma, \zeta)}{d_{\mathcal{W}}(\mathbf{p}, \mathbf{q})} = \frac{\inf \int_0^1 \sqrt{g_{\mathcal{X}}\langle(\mu \circ \pi)'(t), (\mu \circ \pi)'(t)\rangle}\mathrm{dt}}{\inf \int_0^1 \sqrt{g_{\mathcal{W}}\langle\pi'(t), \pi'(t)\rangle}\mathrm{dt}} = \frac{\inf \int_0^1 \sqrt{\mu^*(g_{\mathcal{X}})\langle\pi'(t), \pi'(t)\rangle}\mathrm{dt}}{\inf \int_0^1 \sqrt{g_{\mathcal{W}}\langle\pi'(t), \pi'(t)\rangle}\mathrm{dt}}.$$

We see a bound for this term by Theorem B.3.1.

Theorem B.3.1.

$$tort(\mu) = [m(\mu), M(\mu)],$$

where

$$m(\mu)^2 = \inf_{\mathbf{b} \in \mathcal{W}, v_{\mathbf{b}} \in T_{\mathbf{b}}\mathcal{W}} \frac{\mu^*(g_{\mathcal{X}})\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}{g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}$$

and

$$M(\mu)^2 = \sup_{\mathbf{b} \in \mathcal{W}, v_{\mathbf{b}} \in T_{\mathbf{b}}\mathcal{W}} \frac{\mu^*(g_{\mathcal{X}})\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}{g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}.$$

Proof. Let m_μ and M_μ be two positive numbers such that

$$m_\mu^2 = \inf_{\mathbf{b} \in \mathcal{W}, v_{\mathbf{b}} \in T_{\mathbf{b}}\mathcal{W}} \frac{\mu^*(g_{\mathcal{X}})\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}{g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}$$

and

$$M_\mu^2 = \sup_{\mathbf{b} \in \mathcal{W}, v_{\mathbf{b}} \in T_{\mathbf{b}}\mathcal{W}} \frac{\mu^*(g_{\mathcal{X}})\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}{g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}.$$

Given $\mathbf{p}, \mathbf{q} \in \mathcal{W}$, where $\mu(\mathbf{p}) = \gamma$, $\mu(\mathbf{q}) = \zeta$, for any path π connecting $\mathbf{p}$ and $\mathbf{q}$, we have an estimation

$$\begin{aligned}
\ell(\mu \circ \pi) &= \int_0^1 \sqrt{\mu^*(g_{\mathcal{X}})\langle\pi'(t), \pi'(t)\rangle}\mathrm{dt} \\
&\leq \int_0^1 \sqrt{M_\mu^2 g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}\mathrm{dt} \\
&= M_\mu \int_0^1 \sqrt{g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}\mathrm{dt} \\
&= M_\mu \ell(\pi).
\end{aligned}$$

Similarly, $\ell(\mu \circ \pi) \geq m_\mu \ell(\pi)$.

For any $\mathbf{p}, \mathbf{q} \in \mathcal{W}$, for any $\pi : [0,1] \to \mathcal{W}, \pi(0) = \mathbf{p}, \pi(1) = \mathbf{q}$, there is $\mu \circ \pi : [0,1] \to \mathcal{W}, \mu \circ \pi(0) = \gamma, \mu \circ \pi(1) = \zeta$, such that $\frac{\ell(\mu \circ \pi)}{\ell(\pi)} \leq M(\mu)$, which implies that

$$\frac{\inf_\pi \ell(\mu \circ \pi)}{\inf_\pi \ell(\pi)} \leq M_\mu,$$

and

$$\frac{d_{\mathcal{X}}(\gamma, \zeta)}{d_{\mathcal{W}}(\mathbf{p}, \mathbf{q})} \leq M_\mu.$$

Moreover, since μ is a locally homeomorphism, for the shortest $\pi : [0,1] \to \mathcal{X}, \pi(0) = \gamma, \pi(1) = \zeta$, $\mu^{-1} \circ \pi$ can be placed in a branch of $\mathrm{dom}(\mu)$. So $\mu^{-1} \circ \pi$ exists and

$$\ell(\pi) \leq \frac{1}{m_\mu} \ell(\mu^{-1} \circ \pi).$$

Therefore,

$$\frac{\inf_\pi \ell(\mu^{-1} \circ \pi)}{\inf_\pi \ell(\pi)} \leq \frac{1}{m_\mu},$$

and

$$\frac{d_{\mathcal{X}}(\gamma, \zeta)}{d_{\mathcal{W}}(\mathbf{p}, \mathbf{q})} \geq m_\mu.$$

Hence $m(\mu) \geq m_\mu$ and $M(\mu) \leq M_\mu$.

To prove the equality, we observe that for any $\varepsilon > 0$, there is $\mathbf{b} \in \mathcal{W}$ and $v_{\mathbf{b}} \in T_{\mathbf{b}}\mathcal{W}$, such that

$$\frac{\mu^*(g_{\mathcal{X}})\langle v_{\mathbf{b}}, v_{\mathbf{b}} \rangle}{g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}} \rangle} > M_\mu^2 - 2\varepsilon$$

or

$$\frac{g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}} \rangle}{\mu^*(g_{\mathcal{X}})\langle v_{\mathbf{b}}, v_{\mathbf{b}} \rangle} > \frac{1}{m_\mu^2} - 2\varepsilon.$$

We suppose the former, the latter is similar. Let π be a flow starting at $\mathbf{b}$ with $\pi'(0) = v_{\mathbf{b}}$, we define a sequence of paths $\pi_i(t) = \pi(it)$ for $i = 1, 2, 3, \ldots$. Since $g_{\mathcal{W}}$ and $g_{\mathcal{X}}$ are continuous, there is an open neighborhood $\mathcal{U}_{\mathbf{b}}$ of $\mathbf{b}$ such that for any $\mathbf{p} \in \mathcal{U}_{\mathbf{b}}$,

$$\frac{\mu^*(g_{\mathcal{X}})\langle v_{\mathbf{b}}, v_{\mathbf{b}} \rangle}{g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}} \rangle} > M_\mu^2 - \varepsilon.$$

Then when i is large enough, there is $i \in \mathbb{N}$ such that $\pi_i \subseteq \mathcal{U}_{\mathbf{b}}$, and we have

$$\frac{\ell(\mu \circ \pi)}{\ell(\pi)} > \sqrt{M_\mu^2 - \varepsilon}.$$

This is true for any $\varepsilon > 0$, therefore the bound M_μ is strict, and

$$M(\mu) = M_\mu.$$

$\square$

Corollary B.3.2.

$$tort(\mu) = [m(\mu), M(\mu)],$$

where

$$m(\mu)^2 = \inf_{\mathbf{b} \in \mathcal{W}, g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle = 1} \frac{\mu^*(g_{\mathcal{X}})\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}{g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}$$

and

$$M(\mu)^2 = \sup_{\mathbf{b} \in \mathcal{W}, g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle = 1} \frac{\mu^*(g_{\mathcal{X}})\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}{g_{\mathcal{W}}\langle v_{\mathbf{b}}, v_{\mathbf{b}}\rangle}.$$

B.4 Distortion Bounds for a General Class

This section shows a theorem that is a general version of distortion bounds for representations μ_n.

Theorem B.4.1. *The range of distortion of representation*

$$\mu_n : \partial[-1,1]^{n+1} \to S^n$$
$$\mathbf{x} \mapsto \frac{\mathbf{x}}{|\mathbf{x}|},$$

where $|\mathbf{x}| = \sqrt{x_i^2}$ *is* $[\frac{1}{n+1}, 1]$.

Proof. We simply write $B_0 = \partial[-1,1]^{n+1}$ and $\mathcal{X} = S^n$. Suppose that $\mu_n(\mathbf{x}) = (y_i)$ where $\mathbf{x} = (x_j)$ for $i, j = 0, \ldots, n$. When $i = j$,

$$\frac{\partial y_i}{\partial x_i} = \frac{|\mathbf{x}|^2 - x_i^2}{|\mathbf{x}|^3};$$

and when $i \neq j$,

$$\frac{\partial y_i}{\partial x_j} = -\frac{x_i x_j}{|\mathbf{x}|^3}.$$

For any $v_{\mathbf{x}} \in T_{\mathbf{x}} B_0$ such that $g_{\mathcal{W}}\langle v_{\mathbf{x}}, v_{\mathbf{x}}\rangle = 1$, where $g_{\mathcal{W}}$ is the $\mathbb{R}^{n+1}$ metric which implies $1 = g_{\mathcal{W}}\langle v_{\mathbf{x}}, v_{\mathbf{x}}\rangle = \sum_i \mathrm{dx}_i^2$,

$$\begin{aligned}
\mu_n^*\langle v_{\mathbf{x}}, v_{\mathbf{x}}\rangle &= \sum_i \mathrm{dy}_i^2 \\
&= \sum_i \frac{\partial y_i}{\partial x_j}\frac{\partial y_i}{\partial x_k}\mathrm{dx}_j\mathrm{dx}_k \\
&= \frac{1}{|\mathbf{x}|^6}\sum_j \left((|\mathbf{x}|^2 - x_j^2)^2 + \sum_{i\neq j} x_i^2 x_j^2\right)\mathrm{dx}_j^2 \\
&- \frac{1}{|\mathbf{x}|^6}\sum_{j\neq k}\left((|\mathbf{x}|^2 - x_j^2)(x_j x_k) + (|\mathbf{x}|^2 - x_k^2)(x_k x_j) - \sum_{i\neq j,k} x_i^2 x_j x_k\right)\mathrm{dx}_j\mathrm{dx}_k \\
&= \frac{1}{|\mathbf{x}|^6}\sum_j \left((|\mathbf{x}|^2 - x_j^2)^2 + (|\mathbf{x}|^2 - x_j^2)x_j^2\right)\mathrm{dx}_j^2 \\
&- \frac{1}{|\mathbf{x}|^6}\sum_{j\neq k}\left((|\mathbf{x}|^2 - x_j^2) + (|\mathbf{x}|^2 - x_k^2) - (|\mathbf{x}|^2 - x_j^2 - x_k^2)\right)x_j x_k \mathrm{dx}_j\mathrm{dx}_k \\
&= \frac{1}{|\mathbf{x}|^4}\sum_j \left((|\mathbf{x}|^2 - x_j^2)\mathrm{dx}_j^2 - \sum_{j\neq k} x_j x_k \mathrm{dx}_j\mathrm{dx}_k\right) \\
&= \frac{1}{|\mathbf{x}|^4}\sum_j \left(\sum_{k\neq j} x_k^2\mathrm{dx}_j^2 - \sum_{j\neq k} x_j x_k \mathrm{dx}_j\mathrm{dx}_k\right).
\end{aligned}$$

We consider the facet with $x_0 = 1$ in B_0, and estimate the sum by the following two methods:

1.

$$\begin{aligned}
\frac{1}{|\mathbf{x}|^4}\sum_j \left(\sum_{k\neq j} x_k^2\mathrm{dx}_j^2 - \sum_{j\neq k} x_j x_k \mathrm{dx}_j\mathrm{dx}_k\right) &= \frac{1}{|\mathbf{x}|^4}\sum_k \left(x_k^2(1-\mathrm{dx}_k^2) - \sum_{j\neq k} x_j x_k \mathrm{dx}_j\mathrm{dx}_k\right) \\
&= \frac{1}{|\mathbf{x}|^4}\left[\sum_k x_k^2 - \sum_k \left((x_k\mathrm{dx}_k)^2 + \sum_{j\neq k} x_j x_k \mathrm{dx}_j\mathrm{dx}_k\right)\right] \\
&= \frac{1}{|\mathbf{x}|^4}\left[|\mathbf{x}|^2 - \left(\sum_k x_k \mathrm{dx}_k\right)^2\right] \\
&\leq \frac{1}{|\mathbf{x}|^2}
\end{aligned}$$

$$\leq 1,$$

where the maximum reaches when $x_i = 0$ for all $i > 0$ and any $v_{\mathbf{x}}$ such that $g_{\mathcal{W}}\langle v_{\mathbf{x}}, v_{\mathbf{x}}\rangle = 1$.

2.

$$\begin{aligned}
\frac{1}{|\mathbf{x}|^4}\sum_j \left(\sum_{k\neq j} x_k^2 \mathrm{dx}_j^2 - \sum_{j\neq k} x_j x_k \mathrm{dx}_j \mathrm{dx}_k\right) &= \frac{1}{|\mathbf{x}|^4}\sum_j \sum_{k<j} (x_j \mathrm{dx}_k - x_k \mathrm{dx}_j)^2 \\
&= \frac{1}{|\mathbf{x}|^4}\sum_{0<j}\left(\sum_{0<k<j} (x_j \mathrm{dx}_k - x_k \mathrm{dx}_j)^2 + \mathrm{dx}_j^2\right) \\
&\geq \frac{1}{|\mathbf{x}|^4} \\
&\geq \frac{1}{(n+1)^2},
\end{aligned}$$

where the minimum reaches when $x_i = 1$ for all i and $\mathrm{dx}_i = \frac{1}{\sqrt{n}}$ for all $i > 0$.

By Corollary B.3.2, the distortion of $\mathrm{tort}(\mu_n) = [\frac{1}{n+1}, 1]$. □

B.5 Distortion Bound for Diffeomorphism from $\mathbb{RP}^3$ to $SO(3)$

This section shows the proof of $\rho(\cos\theta + v\sin\theta) = R(v, 2\theta)$.

Theorem B.5.1. *Let $r = \cos\theta + v\sin\theta$ where $v \in \mathbb{H}_p$ and $\theta \in \mathbb{R}$. For any $u \in \mathbb{H}_p$, the map $\rho(r) : u \mapsto rur^{-1}$ is a rotation in $\mathbb{H}_p = \mathbb{R}^3$ such that its axis is v and the rotation angle is 2θ [19].*

Proof. A fact for $v \in \mathbb{H}_p$ is $v^2 = -1$. When $u = v$,

$$\begin{aligned}
\rho(r)(v) &= rvr^{-1} \\
&= ru\bar{r} \\
&= (\cos\theta + v\sin\theta)v(\cos\theta - v\sin\theta) \\
&= (v\cos\theta - \sin\theta)(\cos\theta - v\sin\theta) \\
&= v(\cos^2\theta + \sin^2\theta) - (1 + v^2)\sin\theta\cos\theta
\end{aligned}$$

$$= v$$

Thus $\rho(r)$ fix v.

For any $u \in \mathbb{H}_p = \mathbb{R}^3$ such that $u \cdot v = 0$, define $w = vu$ and we have $w = -(u \cdot v) + u \times v = u \times v$ in $\mathbb{R}^3$. $\{v, u, w\}$ now forms a frame in $\mathbb{R}^3$, and any $q \in\in \mathbb{R}^3$ is a linear combination of $\{v, u, w\}$. We observe that

$$\begin{aligned}
\rho(r)(u) &= rur^{-1} \\
&= ru\overline{r} \\
&= (\cos\theta + v\sin\theta)u(\cos\theta - v\sin\theta) \\
&= (u\cos\theta + vu\sin\theta)(\cos\theta - v\sin\theta) \\
&= u\cos^2\theta + vu\sin\theta\cos\theta - uv\sin\theta\cos\theta - vuv\sin^2\theta \\
&= u\cos^2\theta + 2vu\sin\theta\cos\theta + v^2u\sin^2\theta \\
&= u(\cos^2\theta - \sin^2\theta) + 2w\sin\theta\cos\theta \\
&= u\cos 2\theta + w\sin 2\theta.
\end{aligned}$$

and similarly, $\rho(r)(w) = -u\sin 2\theta + w\cos 2\theta$. Therefore $\rho(r)$ rotates the frame $\{v, u, w\}$ based on v by 2θ degree, which proves that $\rho(r) = R(v, 2\theta)$. $\square$

Appendix C

Exact Footprints for Line Segments

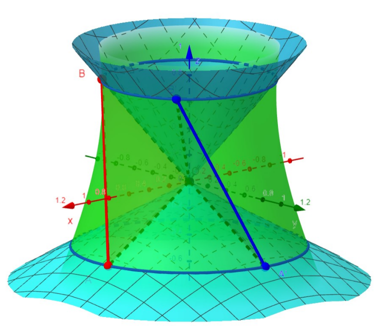
Given $\mathbf{p}, \mathbf{q} \in B^r \in \square\mathcal{W}^r$, the motion on $\overline{\mathbf{pq}}$ is a pure rotation with a fixed axis. For simplicity, we build a local frame such that the rotation axis is the z-axis. For each $\mathbf{b} \in \overline{\mathbf{pq}}$, the coordinates of $\mathcal{A}$ and $\mathcal{B}$ under the local frame are $(x_{\mathcal{A}}, y_{\mathcal{A}}, z_{\mathcal{A}})$ and $(x_{\mathcal{B}}, y_{\mathcal{B}}, z_{\mathcal{B}})$ respectively. Note that $z_{\mathcal{A}}$ and $z_{\mathcal{B}}$ will not change during the rotations. $\mathrm{Fp}(\overline{\mathbf{pq}})$ are classified based on different relations between $z_{\mathcal{A}}$, $z_{\mathcal{B}}$ and 0:

Example C.0.1. When $z_{\mathcal{A}} < 0 < z_{\mathcal{B}}$, $\mathrm{Fp}(\overline{\mathbf{pq}})$ is a subset of

$$\begin{cases} z \geq z_{\mathcal{A}}\sqrt{\frac{x^2+y^2}{x_{\mathcal{A}}^2+y_{\mathcal{A}}^2}}, \\ z \leq z_{\mathcal{B}}\sqrt{\frac{x^2+y^2}{x_{\mathcal{B}}^2+y_{\mathcal{B}}^2}}, \\ \frac{x^2+y^2}{(x_{\mathcal{B}}-x_{\mathcal{A}})^2+(y_{\mathcal{B}}-y_{\mathcal{A}})^2} - \frac{(z-z_{\mathcal{A}}+t)^2}{(z_{\mathcal{B}}-z_{\mathcal{A}})^2} \leq \frac{x_{\mathcal{A}}^2+y_{\mathcal{A}}^2}{(x_{\mathcal{B}}-x_{\mathcal{A}})^2+(y_{\mathcal{B}}-y_{\mathcal{A}})^2} - \frac{t^2}{(z_{\mathcal{B}}-z_{\mathcal{A}})^2}, \end{cases}$$

where

$$t = (z_{\mathcal{B}} - z_{\mathcal{A}})\frac{(x_{\mathcal{B}} - x_{\mathcal{A}})x_{\mathcal{A}} + (y_{\mathcal{B}} - y_{\mathcal{A}})y_{\mathcal{A}}}{(x_{\mathcal{B}} - x_{\mathcal{A}})^2 + (y_{\mathcal{B}} - y_{\mathcal{A}})^2}.$$



When $0 < z_{\mathcal{A}} < z_{\mathcal{B}}$, $\mathrm{Fp}(\overline{\mathbf{pq}})$ is a subset of

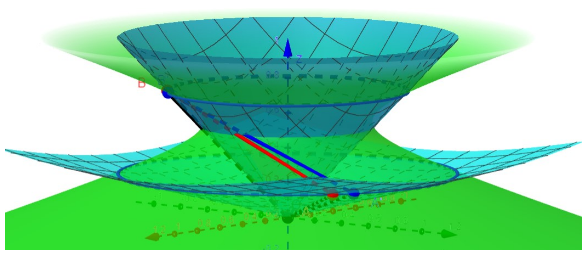
$$\begin{cases} z \geq z_{\mathcal{A}}\sqrt{\frac{x^2+y^2}{x_{\mathcal{A}}^2+y_{\mathcal{A}}^2}} & , \\ z \leq z_{\mathcal{B}}\sqrt{\frac{x^2+y^2}{x_{\mathcal{B}}^2+y_{\mathcal{B}}^2}} & \text{when } z \leq z_\tau, \\ \frac{x^2+y^2}{(x_{\mathcal{B}}-x_{\mathcal{A}})^2+(y_{\mathcal{B}}-y_{\mathcal{A}})^2} - \frac{(z-z_{\mathcal{A}}+t)^2}{(z_{\mathcal{B}}-z_{\mathcal{A}})^2} \leq \frac{x_{\mathcal{A}}^2+y_{\mathcal{A}}^2}{(x_{\mathcal{B}}-x_{\mathcal{A}})^2+(y_{\mathcal{B}}-y_{\mathcal{A}})^2} - \frac{t^2}{(z_{\mathcal{B}}-z_{\mathcal{A}})^2} & \text{when } z \leq z_\tau, \\ z \geq z_{\mathcal{B}}\sqrt{\frac{x^2+y^2}{x_{\mathcal{B}}^2+y_{\mathcal{B}}^2}} & \text{when } z > z_\tau, \\ \frac{x^2+y^2}{(x_{\mathcal{B}}-x_{\mathcal{A}})^2+(y_{\mathcal{B}}-y_{\mathcal{A}})^2} - \frac{(z-z_{\mathcal{A}}+t)^2}{(z_{\mathcal{B}}-z_{\mathcal{A}})^2} \geq \frac{x_{\mathcal{A}}^2+y_{\mathcal{A}}^2}{(x_{\mathcal{B}}-x_{\mathcal{A}})^2+(y_{\mathcal{B}}-y_{\mathcal{A}})^2} - \frac{t^2}{(z_{\mathcal{B}}-z_{\mathcal{A}})^2} & \text{when } z > z_\tau, \end{cases}$$

where z_τ satisfies

$$\left(\frac{x_{\mathcal{B}}^2+y_{\mathcal{B}}^2}{z_{\mathcal{B}}^2\left[(x_{\mathcal{A}}-x_{\mathcal{B}})^2+(y_{\mathcal{A}}-y_{\mathcal{B}})^2\right]} - \frac{1}{(z_{\mathcal{A}}-z_{\mathcal{B}})^2}\right) z = \frac{2t'-z_{\mathcal{B}}}{(z_{\mathcal{A}}-z_{\mathcal{B}})^2} - \frac{z_{\mathcal{B}}(x_{\mathcal{B}}^2+y_{\mathcal{B}}^2)}{z_{\mathcal{B}}^2\left[(x_{\mathcal{A}}-x_{\mathcal{B}})^2+(y_{\mathcal{A}}-y_{\mathcal{B}})^2\right]},$$

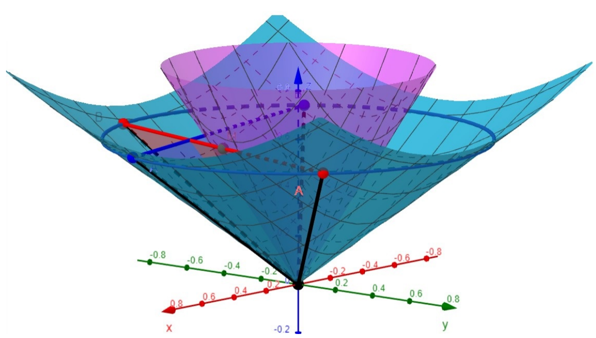
where

$$t' = (z_{\mathcal{A}}-z_{\mathcal{B}})\frac{(x_{\mathcal{A}}-x_{\mathcal{B}})x_{\mathcal{B}}+(y_{\mathcal{A}}-y_{\mathcal{B}})y_{\mathcal{B}}}{(x_{\mathcal{A}}-x_{\mathcal{B}})^2+(y_{\mathcal{A}}-y_{\mathcal{B}})^2}.$$



When $0 < z_{\mathcal{A}} = z_{\mathcal{B}}$, $F_p(\triangle AOB)$ is

$$\begin{cases} z \geq z_{\mathcal{A}}\sqrt{\frac{x^2+y^2}{x_{\mathcal{A}}^2+y_{\mathcal{A}}^2}} = z_{\mathcal{B}}\sqrt{\frac{x^2+y^2}{x_{\mathcal{B}}^2+y_{\mathcal{B}}^2}}, \\ z \leq (z_{\mathcal{A}}+z_{\mathcal{B}})\sqrt{\frac{x^2+y^2}{(x_{\mathcal{A}}+x_{\mathcal{B}})^2+(y_{\mathcal{A}}+y_{\mathcal{B}})^2}}. \end{cases}$$



Appendix D

Semi-Algebraic Expressions, Linear Approximations and Solid Method

D.1 Semi-algebraic Expressions for Π_1 Components in $\widetilde{\mathrm{Fp}}(B)$

Example D.1.1. Let $\mathbf{u}_{\mathcal{AB}}(B) = \frac{\mathrm{m}_{\mathcal{B}}(B)-\mathrm{m}_{\mathcal{A}}(B)}{|\mathrm{m}_{\mathcal{B}}(B)-\mathrm{m}_{\mathcal{A}}(B)|}$.

The Cylinder(B) is the semi-algebraic set of $\mathbf{x} \in \mathbb{R}^3$ with constraints:

$$\begin{cases} \left|(\mathbf{x} - \mathrm{m}_{\mathcal{A}}(B)) \times \mathbf{u}_{\mathcal{AB}}(B)\right| \leq \mathrm{d}(B) \\ (\mathbf{x} - \mathrm{m}_{\mathcal{A}}(B)) \cdot \mathbf{u}_{\mathcal{AB}}(B) \geq 0 \\ (\mathbf{x} - \mathrm{m}_{\mathcal{B}}(B)) \cdot \mathbf{u}_{\mathcal{AB}}(B) \leq 0 \end{cases}.$$

See Figure D.1.

Example D.1.2. Let $\mathbf{p}_{\mathcal{A}}(B) = \frac{\mathrm{d}(B)\mathrm{m}(B^t)-\mathrm{r}(B)\mathrm{m}_{\mathcal{A}}(B)}{\mathrm{d}(B)-\mathrm{r}(B)}$, $\mathbf{u}_{\mathcal{OA}}(B) = \frac{\mathrm{m}_{\mathcal{A}}(B)-\mathrm{m}_{\mathcal{O}}(B)}{|\mathrm{m}_{\mathcal{A}}(B)-\mathrm{m}_{\mathcal{O}}(B)|}$, $Q_{\mathcal{A}}(B) = [\mathbf{u}_{\mathcal{OA}}(B)]^T[\mathbf{u}_{\mathcal{OA}}(B)] - \frac{\mathrm{r}(B)^2}{|\mathbf{p}_{\mathcal{A}}(B)|^2} I$, where $[\cdot]$ is the Lie-bracket of $\mathbb{R}^3$ vectors, I is 3×3 identity matrix.

The Frustum$_{\mathcal{A}}(B)$ is the semi-algebraic set of $\mathbf{x} \in \mathbb{R}^3$ with constraints:

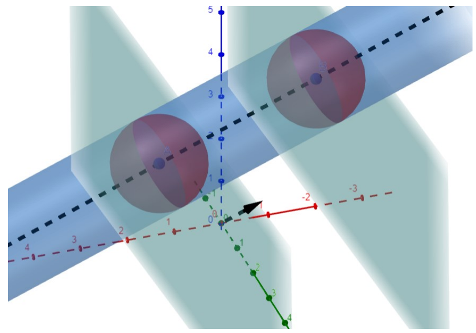


Figure D.1: Construction of Π_1 expression for the cylinder.

$$\begin{cases} \mathbf{x}^T Q_{\mathcal{A}}(B)\mathbf{x} - 2\mathbf{p}_{\mathcal{A}}(B) Q_{\mathcal{A}}(B)\mathbf{x} + \mathbf{p}_{\mathcal{A}}(B)^T Q_{\mathcal{A}}(B)\mathbf{p}_{\mathcal{A}}(B) \leq 0 \\ (\mathbf{x} - \mathrm{m}(B^t)) \cdot \mathbf{u}_{\mathcal{O}\mathcal{A}}(B) \leq -\frac{\mathrm{r}(B)^2}{|\mathbf{p}_{\mathcal{A}}(B)|} \\ (\mathbf{x} - \mathrm{m}_{\mathcal{A}}(B)) \cdot \mathbf{u}_{\mathcal{O}\mathcal{A}}(B) \leq -\frac{\mathrm{d}(B)\mathrm{r}(B)}{|\mathbf{p}_{\mathcal{A}}(B)|} \end{cases}.$$

See Figure D.2.

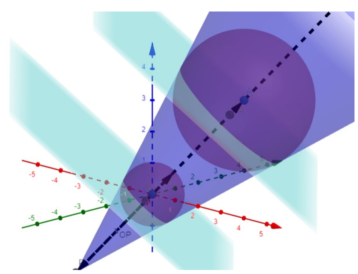


Figure D.2: Construction of Π_1 expression for the frustum.

Example D.1.3. The pyramid is the convex hull of 3 parallel line segments. Their end points are

$$\mathrm{S}_{\mathcal{A}}(B) \cap \mathrm{Cylinder}(B) \cap \mathrm{Frustum}_{\mathcal{A}}(B),$$

$$\mathrm{S}_{\mathcal{B}}(B) \cap \mathrm{Cylinder}(B) \cap \mathrm{Frustum}_{\mathcal{B}}(B),$$

$$\mathrm{S}_{\mathcal{O}}(B) \cap \mathrm{Frustum}_{\mathcal{A}}(B) \cap \mathrm{Frustum}_{\mathcal{B}}(B)$$

respectively.

Let $\mathbf{u}_{\mathcal{O}\mathcal{A}}(B) = \frac{\mathrm{m}_{\mathcal{A}}(B) - \mathrm{m}_{\mathcal{O}}(B)}{|\mathrm{m}_{\mathcal{A}}(B) - \mathrm{m}_{\mathcal{O}}(B)|}$, $\mathbf{u}_{\mathcal{O}\mathcal{B}}(B) = \frac{\mathrm{m}_{\mathcal{B}}(B) - \mathrm{m}_{\mathcal{O}}(B)}{|\mathrm{m}_{\mathcal{B}}(B) - \mathrm{m}_{\mathcal{O}}(B)|}$, and $\mathbf{u}_{\mathcal{A}\mathcal{B}}(B) = \frac{\mathrm{m}_{\mathcal{B}}(B) - \mathrm{m}_{\mathcal{A}}(B)}{|\mathrm{m}_{\mathcal{B}}(B) - \mathrm{m}_{\mathcal{A}}(B)|}$.

The constraints for $\mathbf{x} \in \mathrm{S}_{\mathcal{A}}(B) \cap \mathrm{Cylinder}(B) \cap \mathrm{Frustum}_{\mathcal{A}}(B)$ are

$$\begin{cases} \mathbf{x} \cdot \mathbf{u}_{\mathcal{AB}}(B) = 0 \\ \mathbf{x} \cdot \mathbf{u}_{\mathcal{OA}}(B) = -\frac{\mathrm{d}(B)\mathrm{r}(B)}{|\mathbf{p}_{\mathcal{A}}(B)|} \\ |\mathbf{x} - \mathrm{m}_{\mathcal{A}}(B)| = \mathrm{d}(B) \end{cases}.$$

Similar for $\mathbf{x} \in \mathrm{S}_{\mathcal{B}}(B) \cap \mathrm{Cylinder}(B) \cap \mathrm{Frustum}_{\mathcal{B}}(B)$.

The constraints for $\mathrm{S}_{\mathcal{O}}(B) \cap \mathrm{Frustum}_{\mathcal{A}}(B) \cap \mathrm{Frustum}_{\mathcal{B}}(B)$ are

$$\begin{cases} \mathbf{x} \cdot \mathbf{u}_{\mathcal{OA}}(B) = -\frac{\mathrm{d}(B)\mathrm{r}(B)}{|\mathbf{p}_{\mathcal{A}}(B)|} \\ \mathbf{x} \cdot \mathbf{u}_{\mathcal{OB}}(B) = -\frac{\mathrm{d}(B)\mathrm{r}(B)}{|\mathbf{p}_{\mathcal{B}}(B)|} \\ |\mathbf{x} - \mathrm{m}_{\mathcal{O}}(B)| = \mathrm{r}(B) \end{cases}.$$

The pyramid is the convex hull of the 6 end points. See Figure D.4.

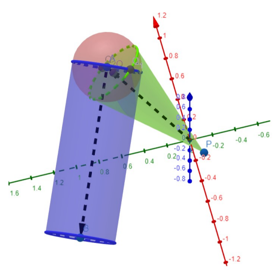


Figure D.3: Construction of corners of the pyramid.

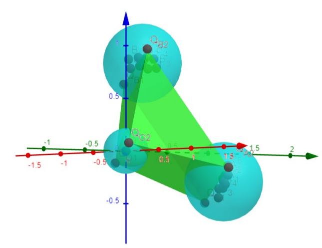


Figure D.4: Construction of the pyramid.

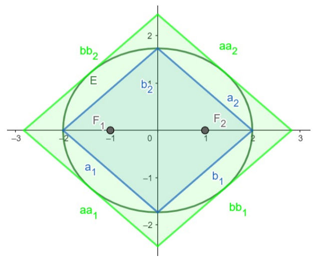
D.2 Linear Approximation for Planar Semi-Quadratic Sets

The linear approximation for maintaining R in the double loop is given by the following theory. The key idea is the linear approximation scheme. Since our elementary sets are semi-algebraic sets with degree at most 2, their intersections with a parametric plane can be classified as one of ellipse, parabola and hyperbola. The linear approximation scheme is applied to these primitives (Π_1 sets).

Definition D.2.1. Given a set $\mathcal{S}$ of primitives, a **linear approximation scheme** is the set of two operators $\underline{\cdot}$ and $\overline{\cdot}$ such that for each $S_p \in \mathcal{S}$, $\underline{S_p}$ and $\overline{S_p}$ are two linear semi-algebraic sets and

$$\underline{S_p} \subseteq S_p \subseteq \overline{S_p}.$$

Example D.2.2. When S_p is an ellipse, we design the following approximations:



First, we first find the longer axis and shorter axis of the ellipse, then find the endpoints of the two axes. The four end points form a diamond which is the inscribing approximation.

Second, we find the tangents of the ellipse that are parallel to the edges of the inscribed approximation. The four tangents form another diamond which is the circumscribing approximation.

Given the structure of the two approximations, we have the following geometric fact:

Proposition D.2.3. *The circumscribing and inscribing approximations above of the ellipse are homothetic*[1] *with a ratio* $\sqrt{2}$.

Proof. Let the ellipse to be

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

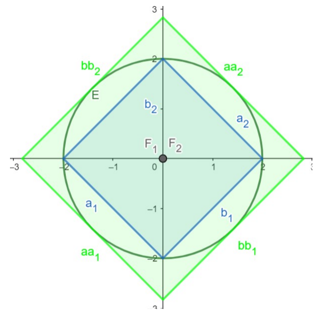
We can apply an affine transformation τ on $\mathbb{R}^2$ by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \frac{1}{a} & 0 \\ 0 & \frac{1}{b} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Then τ maps the ellipse to a circle and keeps the inscribing relationship between ellipse and its two approximations. Hence, the image of τ gives the unit circle, and the image of the inscribing approximation

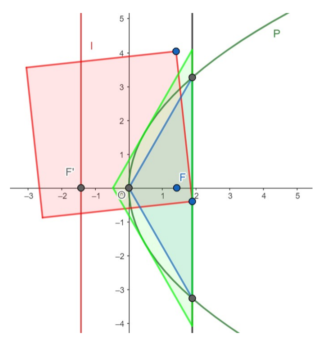
[1] We refer to the homothety between tiles defined in Chapter 2.

of the ellipse is the inscribing approximation of the unit circle, and the image of the circumscribing approximation of the ellipse is the circumscribing approximation of the unit circle.



From the unit circle, we see that the two approximations of the unit circle are homothetic with ratio $\sqrt{2}$. Since the affine transformation keeps homotheticity and the ratio, we know that the two approximations of the ellipse are still homothetic with ratio $\sqrt{2}$. $\square$

Example D.2.4. When S_p is a parabola, we need to involve the universal boundary and we design the following approximations:



First, we find the vertex of the universal boundary that projects the maximum onto the symmetric axis of the parabola with positive direction to be the opening direction of the parabola. Then, we set a perpendicular line through the vertex to be the finite boundary of the infinite parabola.

Second, we take the triangle formed by the vertex of the parabola and the two intersections between the finite boundary with the parabola as the inscribed approximation of S_p.

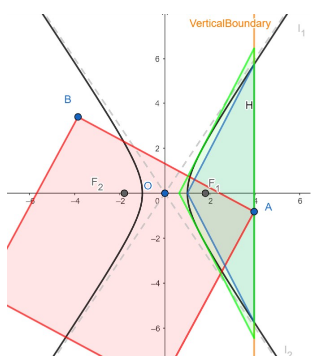
Third, we find two tangent lines on the parabola that are parallel to the two edges of $\underline{S_p}$. The two tangent lines and the finite boundary together form another triangle which we take it as the circumscribing approximation of S_p.

Given the structure of the two approximations, we have the following geometry fact:

Proposition D.2.5. *The circumscribing and inscribing approximations above of the parabola are homothetic with ratio* $\frac{5}{4}$.

Proof. Let the parabola be $P : y^2 = 2px$, where p is the distance between the focal and the directrix. Let the finite boundary given by the vertex of the universal boundary be $l : x = c$. By a quick calculation, the two intersections between P and l are $(c, \pm\sqrt{2pc})$. Hence the "slopes" of the two non-vertical edges are $\frac{dx}{dy} = \pm\sqrt{\frac{c}{2p}}$. The points on the parabola that has the same "slopes" should satisfies $\pm\sqrt{\frac{c}{2p}} = \frac{dx}{dy} = \frac{y}{p}$. Hence the two tangent points are $(\frac{c}{4}, \pm\sqrt{\frac{pc}{2}})$. Then, the equation of the two tangent lines are $(x - \frac{c}{4}) = \pm\sqrt{\frac{c}{2p}}(y \mp \sqrt{\frac{pc}{2}})$. Taking y to be 0 in the equation, we have $x = \frac{c}{4} - \sqrt{\frac{c}{2p}}\sqrt{\frac{pc}{2}} = -\frac{c}{4}$. Hence the left vertex of the circumscribing approximation is $(-\frac{c}{4}, 0)$. Therefore, the height of the inscribing approximation and circumscribing approximation based on l are c and $\frac{5}{4}c$ respectively. Since the edges between the two approximations are always parallel, the two triangles have to be homothetic. The ratio between the height is $\frac{5}{4}$ which is also the ratio of the homthety. $\square$

Example D.2.6. When S_p is a hyperbola, we refer to one of its branches. We also involve the universal boundary and we design the following approximations:



First, we set the same kind of perpendicular line through a vertex of the universal boundary to be the finite boundary of the infinite hyperbola branch.

Second, we take the triangle formed by the vertex of the hyperbola branch and the two intersections between the finite boundary with the hyperbola branch as the inscribing approximation of S_p.

Third, we find two tangent lines on the hyperbola branch that are parallel to the two edges of $\underline{S_p}$. The two tangent lines and the finite boundary together form another triangle which we take it as the circumscribing approximation of S_p.

Given the structure of the two approximations, we have the following geometry fact:

Proposition D.2.7. *The circumscribing and inscribing approximations above of the hyperbola branch are similar with ratio less than* $\frac{5}{4}$.

Proof. For any hyperbola, we can always apply an affine transformation so that its equation becomes $x^2 - y^2 = 1$. The affine transformations keep the homotheticity and ratio, so we study the case under $x^2 - y^2 = 1$.

Let the vertical boundary be $x = c$. Suppose that the "slope" of the upper non-vertical edge of $\underline{S_p}$ is to be $\frac{dx}{dy} = t$. Then the equation of the edge is $x - 1 = ty$. The two solutions of the equations

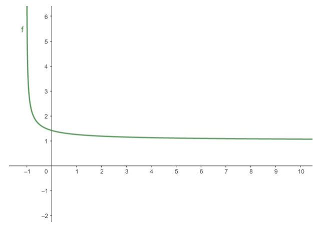
$$\begin{cases} x-1 &= ty \\ x^2-y^2 &= 1 \end{cases}$$

is $(1, 0)$ and $(\frac{1+t^2}{1-t^2}, \frac{2t}{1-t^2})$. Hence, we have the relation $c = \frac{1+t^2}{1-t^2} = \frac{2}{1-t^2} - 1$.

Consider the point on the hyperbola whose tangent is parallel to the edge of $\underline{S_p}$, we will have $\frac{y}{x} = \frac{dx}{dy} = t$ and $x^2 - y^2 = 1$. The solution for the upper edge is $(\frac{1}{\sqrt{1-t^2}}, \frac{t}{\sqrt{1-t^2}})$. Hence the equation of the upper non-vertical edge of $\overline{S_p}$ is $x - \frac{1}{\sqrt{1-t^2}} = t(y - \frac{t}{\sqrt{1-t^2}})$. Taking $y = 0$, we have the intersection on x-axis to be $(\sqrt{1-t^2}, 0)$. Substitute t with c. The intersection is $(\sqrt{\frac{2}{1+c}}, 0)$.

Since the edges between $\underline{S_p}$ and $\overline{S_p}$ are parallel respectively, the two triangles have to be homothetic. The ratio is the ratio of heights taking $x = c$ as the relative center, which is $\frac{c-\sqrt{\frac{2}{1+c}}}{c-1}$.

Let's study the function $f(c) = \frac{c-\sqrt{\frac{2}{1+c}}}{c-1}$ on $c \in (1, +\infty)$.



We set $g(\omega) = f(c)$ with $\omega = \sqrt{\frac{2}{1+c}}$, then $c = \frac{2}{\omega^2} - 1$. By calculation,

$$\begin{aligned} g(\omega) &= \frac{\frac{2}{\omega^2} - 1 - \omega}{\frac{2}{\omega^2} - 1 - 1} \\ &= \frac{2 - \omega^2 - \omega^3}{2 - 2\omega^2} \\ &= \frac{\omega^2 + 2\omega + 2}{2\omega + 2} \\ &= \frac{1}{2}\left((\omega + 1) + \frac{1}{\omega + 1}\right). \end{aligned}$$

Hence, $g(\omega)$ is monotonic increasing on $(0, +\infty)$. Since $c = \frac{2}{\omega^2} - 1$ is monotonic decreasing on $(0, +\infty)$, and $c = 1$ if and only if $\omega = 1$, we have $f(c)$ is monotonic decreasing on $(1, +\infty)$. Therefore,

$$f(c) < \lim_{c \to 1} f(c) = \lim_{\omega \to 1} g(\omega) = \frac{5}{4}.$$

This proves that the ratio of homothety between $\underline{S_p}$ and $\overline{S_p}$ is always less than $\frac{5}{4}$. This ratio keeps unchanged under affine transformations, so it is applicable to all hyperbolas. $\square$

The validation of inscribe and circumscribing approximations, similar to approximate footprints, can be defined as σ-effectiveness.

Definition D.2.8. Given a set $\mathcal{S}$ of primitives S_p, a fixed approximation scheme $\underline{\cdot}$ and $\bar{\cdot}$ is σ**-effective** for $\sigma \geq 1$, if for each $S_p \in \mathcal{S}$, we have

$$\underline{S_p} \subseteq S_p \subseteq \sigma \underline{S_p}$$

and

$$S_p \subseteq \overline{S_p} \subseteq \sigma^2 \underline{S_p}.$$

Given a set $\mathcal{S}$ of primitives S_p, a sequence of approximation schemes $\underline{\cdot}_k$ and $\overline{\cdot}^k$ is σ-effective for $\sigma \geq 1$, if each of the schemes is σ-effective, and moreover, for all $k \in \mathbb{N}$,

$$\underline{S_p}_k \subseteq \underline{S_p}_{k+1} \subseteq S_p$$

and

$$S_p \subseteq \overline{S_p}^{k+1} \subseteq \overline{S_p}^k,$$

we require that

$$\bigcup_{k=1}^{+\infty} \underline{S_p}_k = S_p$$

and

$$\bigcap_{k=1}^{+\infty} \overline{S_p}^k = S_p.$$

Proposition D.2.9. *A fixed approximation scheme for a set of convex primitives is σ-effective if for each $S_p \in \mathcal{S}$,*

$$\overline{S_p} \subseteq \sigma \underline{S_p}.$$

Proof. Since $\sigma \geq 1$, when the scheme satisfies that for each $S_p \in \mathcal{S}$, $\overline{S_p} \subseteq \sigma \underline{S_p}$, and since S_p is convex, we have

$$\underline{S_p} \subseteq S_p \subseteq \overline{S_p} \subseteq \sigma \underline{S_p} \subseteq \sigma^2 \underline{S_p}.$$

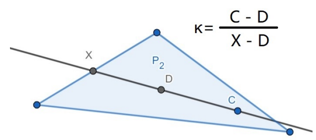
Hence, the scheme is σ-effective. $\square$

A very important property that is used to prove the σ-effective condition is the homothety property. Here we have the following lemma:

Lemma D.2.10. *Suppose that two convex polyhedrons P_1 and P_2 are homothetic with ratio $\tau > 1$ and $P_2 \subseteq P_1$. Let the relative center of the homothety be $\mathbf{c}$ and we denote $\mathbf{d} = m(P_2)$. Define*

$$\kappa = \sup \left\{ r < 0 \middle| \mathbf{d} + \frac{1}{r}(\mathbf{c} - \mathbf{d}) \in P_2 \right\}.$$

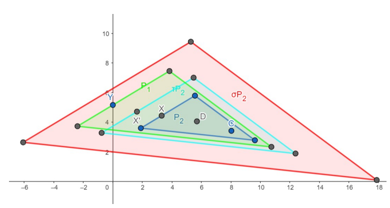
Then for $\sigma = (1-\kappa)\tau + \kappa$, $P_1 \subseteq \sigma P_2$. Especially, if $m(P_1) = m(P_2)$, then for $\sigma = \tau$, $P_1 \subseteq \sigma P_2$.



Proof. Since $P_2 \subseteq P_1$, we have $\mathbf{c} \subseteq P_2$.

For each $\mathbf{y} \in P_1$, since P_1 and P_2 are homothetic, there is $\mathbf{x} \in P_2$ such that $\mathbf{y} - \mathbf{c} = \tau(\mathbf{x} - \mathbf{c})$. Hence we have

$$\begin{aligned}
\mathbf{y} - \mathbf{d} &= (\mathbf{y} - \mathbf{c}) + (\mathbf{c} - \mathbf{d}) \\
&= \tau\mathbf{x} - \tau\mathbf{c} + \mathbf{c} - \mathbf{d} \\
&= \tau(\mathbf{x} - \mathbf{d}) + (1 - \tau)(\mathbf{c} - \mathbf{d}).
\end{aligned}$$



Hence

$$\begin{aligned}
\rho_{P_2}(\mathbf{y}) &= \inf\left\{ r > 0 \middle| \mathbf{d} + \frac{1}{r}(\mathbf{y} - \mathbf{d}) \in P_2 \right\} \\
&= \inf\left\{ r > 0 \middle| \mathbf{d} + \frac{1}{r}(\tau(\mathbf{x} - \mathbf{d}) + (1 - \tau)(\mathbf{c} - \mathbf{d})) \in P_2 \right\} \\
&\leq \inf\left\{ r > 0 \middle| \mathbf{d} + \frac{1}{r}(\tau(\mathbf{x} - \mathbf{d}) + (1 - \tau)\kappa(\mathbf{x} - \mathbf{d})) \in P_2 \right\} \\
&= \inf\left\{ r > 0 \middle| \mathbf{d} + \frac{(1 - \kappa)\tau + \kappa}{r}(\mathbf{x} - \mathbf{d}) \in P_2 \right\} \\
&= (1 - \kappa)\tau + \kappa.
\end{aligned}$$

When $\mathbf{y}$, $\mathbf{c}$ and $\mathbf{d}$ are on the same line, the equality above can be reached. Hence $\sup_{\mathbf{y} \in P_1} \rho_{P_2}(\mathbf{y}) = (1 - \kappa)\tau + \kappa$. Thus when $\sigma = (1 - \kappa)\tau + \kappa$, we have $P_1 \subseteq \sigma P_2$.

When $\mathbf{bc}(P_1) = \mathbf{bc}(P_2)$, since the unique fixed point of homothety is the relative center $\mathbf{c}$, we have $\mathbf{c} = \mathbf{d}$, and therefore, $\kappa = 0$. Hence σ can reach τ. $\square$

As a result, the approximation scheme we defined for ellipse in previous section is $\sqrt{2}$-effective, the approximation scheme we defined for parabola in previous section is $\frac{11}{8}$-effective, the approximation scheme we defined for hyperbola branch in previous section is also $\frac{11}{8}$-effective. By comparison, $\sqrt{2} \approx 1.414 > 1.375 = \frac{11}{8}$, so our approximation scheme for planar quadratic primitives is $\sqrt{2}$-effective in general.

D.3 Solid Method for Delta Robot

Detecting if $\mathrm{Sep}(f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s$ is done by reductions by f. The feature f can be corner, edge or facet. Their corresponding algebraic spans are point, line and plane. We describe the method of answering those queries from point, to line, and to plane.

Lemma D.3.1. *Let f be a corner with the corresponding point be $\mathbf{q}$. Let T be the plane containing $\mathbf{v}$, $\mathbf{c}$ and $\mathbf{q}$. Then, Sep($f, tc(\mathbf{v}, \mathbf{c}, r)^\circ) > s$ if and only if Sep($f, T \cap tc(\mathbf{v}, \mathbf{c}, r)^\circ)) > s$, where $T \cap tc(\mathbf{v}, \mathbf{c}, r)^\circ$ is a union of two intersecting open line segments.*

Proof. Let $\mathbf{p}$ be a point in $\mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ$ such that $(\mathbf{p}, \mathbf{q})$ is an extreme pair, $\mathbf{n_p}$ be the normal vector of the traffic cone at $\mathbf{p}$. By Corollary 5.2.3, $\mathbf{n_p} /\!/ (\mathbf{p} - \mathbf{q})$. As a geometric property of a cone, $\mathbf{n_p}$ is a linear combination of $\mathbf{v} - \mathbf{c}$ and $\mathbf{v} - \mathbf{p}$. Therefore, $\mathbf{p} - \mathbf{q}$ is a linear combination of $\mathbf{v} - \mathbf{c}$ and $\mathbf{v} - \mathbf{p}$, which implies that $\mathbf{p} \in T$. So $\mathrm{Sep}(f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s$ if and only if $\mathrm{Sep}(f, T \cap \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ)) > s$. □

Base on Lemma D.3.1, the process for detecting $\mathrm{Sep}(f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s$ is follow:

$\mathbf{Sep}(f, \mathbf{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s$?

Input: $\mathbf{v}, \mathbf{c} \in \mathcal{Z} = \mathbb{R}^3$, feature $f = \{\mathbf{q}\} \subseteq \mathbb{R}^3$, $r > 0$, $s \geq 0$.

Output: boolean $(\mathrm{Sep}(f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s)$.

$T \leftarrow \mathrm{plane}(\mathbf{v}, \mathbf{c}, \mathbf{q})$.

$l_1 \cup l_2 \leftarrow T \cap \overline{\mathrm{tc}(\mathbf{v}, \mathbf{c}, r)}$.

For $i = 1, 2$

If $\mathrm{Sep}(l_i, f) \leq s$,

Let $\mathbf{p} \in l_i$ such that $|\mathbf{p} - \mathbf{q}| = \mathrm{Sep}(l_i, f)$

If $\mathbf{p} \in \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ$,

return `false`.

return `true`.

Example D.3.2. Given an edge f, checking if $\mathrm{Sep}(f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s$ is generally done by boundary reduction by Theorem 5.2.4. The step $\mathrm{Sep}(\partial f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s$ is recursively done by Lemma D.3.1. The problem is to collect extreme pairs in $\mathrm{ext}(\overline{f}, \overline{\mathrm{tc}(\mathbf{v}, \mathbf{c}, r)}$. To simplify the problem, let us set $\mathbf{v} = (0, 0, 0)$, $\mathbf{c} = (0, 0, -1)$ and $\overline{\mathrm{tc}(\mathbf{v}, \mathbf{c}, r)}$ is CONE $= \{(x, y, z) : x^2 + y^2 - r^2z^2 = 0\}$. General cases can be given after isometry transformations. Suppose that l is parametrized by $\mathbf{q}(t) = \mathbf{q}_0 + \mathbf{u}t$ for $t \in \mathbb{R}$, where $\mathbf{u} = (a, b, c)$. Given a point $\mathbf{p} \in$ CONE, the normal vector of $\mathbf{p}$ is $\mathbf{n_p}$. Suppose that $\mathbf{p} = (x, y, z)$, then $\mathbf{n_p} = (x, y, -r^2z)$. By Corollary 5.2.2, $\mathbf{n_p} \perp \mathbf{u}$. This yields two equations to be solved

$$x^2 + y^2 = r^2z^2 \tag{D.1}$$

$$ax + by = r^2cz \tag{D.2}$$

Let $\kappa := rc$. Multiplying Equation D.1 by κ^2 and subtracting square of Equation D.2, we get a quadratic equation for x and y:

$$(\kappa^2 - a^2)x^2 + (\kappa^2 - b^2)y^2 - 2abxy = 0,$$

which is equivalently:

$$\begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} \kappa^2 - a^2 & -ab \\ -ab & \kappa^2 - b^2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 0.$$

Let

$$A = \begin{pmatrix} \kappa^2 - a^2 & -ab \\ -ab & \kappa^2 - b^2 \end{pmatrix}.$$

Then

$$\det(A) = (\kappa^2 - a^2)(\kappa^2 - b^2) - a^2b^2 = \kappa^2(\kappa^2 - a^2 - b^2).$$

When $\det(A) > 0$, the equation has no real non-trivial ($x = y = 0$ is trivial) solution. Hence $\mathrm{ext}(\mathrm{CONE}, \overline{f}) = \emptyset$.

When $\det(A) \leq 0$, the equation has real non-trivial solutions $(\kappa^2 - a^2)x + (ab \pm \sqrt{-\det(A)})y = 0$, which are one or two planes. We denote the planes by P_1 and P_2. To find the exact $\mathbf{p}$ and $\mathbf{q}$, one only need to notice that $\mathbf{q}_1 = P_1 \cap \overline{f}$ and $\mathbf{q}_2 = P_2 \cap \overline{f}$ ($P_i \cap \overline{f} \neq \emptyset$ since $\mathbf{p} \perp \overline{f}$ and P_i contains the direction of $\mathbf{p}$). Then the corresponding pairs $(\mathbf{p}_j, \mathbf{q}_i)$ are given by Lemma D.3.1.

Lemma D.3.3. *Let f be a facet feature such that $Sep(\partial f, tc(\mathbf{v}, \mathbf{c}, r)) > s$, and $Sep(f, \partial tc(\mathbf{v}, \mathbf{c}, r)) > s$. Then $Sep(f, tc(\mathbf{v}, \mathbf{c}, r)) > s$ if and only if $f \cap \overline{\mathbf{vc}} = \emptyset$.*

Proof. Let $\mathrm{TC} = \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)$, $\mathrm{CONE} = \overline{\mathrm{TC}}$. The necessity of the lemma is obvious. We discuss sufficiency by classification of $\overline{f} \cap \mathrm{CONE}$, where $\overline{f}$ is a plane and CONE is an infinite double cone.

The set $\overline{f} \cap \mathrm{CONE}$ is the intersection between a cone and an infinite double cone, which may be an ellipse, a parabola, a hyperbola, a line, an intersection of two lines, or a point, where the last 3 are called degenerate intersection. When the intersection is degenerate, the apex of the cone $\mathbf{v}$ is always in the intersection, which contradicts that $\mathrm{Sep}(f, \partial\mathrm{TC}) > s$. Therefore, the intersection cannot be degenerated.

Suppose that $\mathrm{ext}(\overline{f}, \mathrm{CONE}) \cap (f \times \mathrm{TC}) \neq \emptyset$, otherwise by Theorem 5.2.4, $\mathrm{Sep}(f, \mathrm{TC}) > s$. The case $\mathrm{ext}(\overline{f}, \mathrm{CONE}) \cap (f \times \mathrm{TC}) \neq \emptyset$ implies that $f \cap \mathrm{TC} \neq \emptyset$. We prove that under this assumption, we always have $f \cap \overline{\mathbf{vc}} \neq \emptyset$.

When $\overline{f} \cap \mathrm{CONE}$ is a parabola or hyperbola, the intersection is unbounded. Under this case, to make $f \cap \mathrm{TC}$ finite, the infinite curve $\overline{f} \cap \mathrm{CONE}$ has to be cut down either by ∂f or by $\partial\mathrm{TC}$. Whichever case contradicts either $\mathrm{Sep}(\partial f, \mathrm{TC}) > s$ or $\mathrm{Sep}(f, \partial\mathrm{TC}) > s$. Hence these two cases are always impossible.

When $\overline{f} \cap \mathrm{CONE}$ is an ellipse. When $\overline{f} \cap \partial\mathrm{TC} \neq \emptyset$, to avoid that $f \cap \mathrm{TC}$ reaches $\partial\mathrm{TC}$, the intersection curve has to be cut down by ∂f. It contradicts $\mathrm{Sep}(\partial f, \mathrm{TC}) > s$. When $\overline{f} \cap \partial\mathrm{TC} = \emptyset$, let $\mathcal{J}$ be the interior of the ellipse disc given by $f \cap \mathrm{TC}$. It is obvious that $f \cap \overline{\mathbf{vc}} = \mathcal{J} \cap \overline{\mathbf{vc}} \neq \emptyset$ since $\overline{\mathbf{vc}}$ is in the interior of the "convex" area bounded by CONE. So in this case, we conclude $f \cap \overline{\mathbf{vc}}$, see Figure D.5 for a reference. The classifications prove the lemma. □

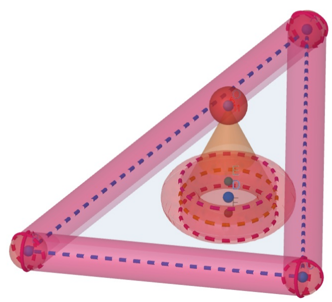


Figure D.5: The case when $f \cap \mathrm{TC} \neq \emptyset$ but $\overline{f} \cap \partial\mathrm{TC} = \emptyset$.

Based on Lemma D.3.3, the process for checking $\mathrm{Sep}(f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)) > s$ for facet f is done by the followings:

$\mathbf{Sep}(f, \mathbf{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s$**?**
Input: $\mathbf{v}, \mathbf{c} \in \mathcal{Z} = \mathbb{R}^3$, feature $f = \triangle\mathbf{abc} \subseteq \mathbb{R}^3$, $r > 0$, $s \geq 0$. Output: boolean $(\mathrm{Sep}(f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) > s)$.
If $(\mathrm{Sep}(\partial f, \mathrm{tc}(\mathbf{v}, \mathbf{c}, r)^\circ) \leq s)$, return `false`. Return $\mathrm{Sep}(f, \overline{\mathbf{vc}}) > 0$.

Appendix E

Explicit Subdivision Methods

E.1 How to Achieve the Inheritance Property without $\widetilde{\mathrm{Fp}}(B) \subseteq \widetilde{\mathrm{Fp}}(\mathbf{parent}(B))$

Example E.1.1. Suppose we have current $\widetilde{\mathrm{Fp}}(B)$ for all $B \in \square\mathcal{W}$. We define a reduced-approximate footprint by induction of $\widehat{\mathrm{Fp}}(B) = \widetilde{\mathrm{Fp}}(B) \cap \widehat{\mathrm{Fp}}(\mathrm{parent}(B))$ for all $B \in \square\mathcal{W}$ other than B_0, where $\widehat{\mathrm{Fp}}(B_0) = \widetilde{\mathrm{Fp}}(B_0)$. This $\widehat{\mathrm{Fp}}$ is still σ-effective since by induction

$$\mathrm{Fp}(B) = \mathrm{Fp}(B) \cap \mathrm{Fp}(\mathrm{parent}(B)) \subseteq \widetilde{\mathrm{Fp}}(B) \cap \widehat{\mathrm{Fp}}(\mathrm{parent}(B)) = \widehat{\mathrm{Fp}}(B)$$

and

$$\widehat{\mathrm{Fp}}(B) \subseteq \widetilde{\mathrm{Fp}}(B) \subseteq \mathrm{Fp}(\sigma B).$$

Now we define $\widehat{\phi}(B)$ be the feature set of $\widehat{\mathrm{Fp}}(B)$, i.e.,

$$\widehat{\phi}(B) = \{f \in \Phi(\Omega) : f \cap \widehat{\mathrm{Fp}}(B) \neq \emptyset\}.$$

This feature set surely satisfies the inheritance property since $\widehat{\mathrm{Fp}}(B) \subseteq \widehat{\mathrm{Fp}}(\mathrm{parent}(B))$. Moreover, by definition of $\widetilde{\phi}(B)$, which is constructed from the inheritance property, it is the same with $\widehat{\phi}(B)$ for each B in $\mathcal{T}(B_0)$. Therefore, we can view the $\widetilde{\phi}(B) = \widehat{\phi}(B)$ as the feature set for $\widehat{\mathrm{Fp}}(B)$, which provides an σ-effective SSS planner.

E.2 Table of Indicator Flips

Example E.2.1. We demonstrate the explicit process to determine the relationships between child indicators across the boundary of C_i for $i = 1, 2, 3, 4$. We use C_1 and C_2 for example. For the box indicated by $(*, \bar{1}, \bar{1}, 1)$ in C_1, one of its corners is $(-1, -1, -1, +1) \in B_0$. This point is a corner of $(\bar{1}, *, \bar{1}, 1)$ in C_2. So $(*, \bar{1}, \bar{1}, 1)$ and $(\bar{1}, *, \bar{1}, 1)$ share the same corner, which implies they are adjacent. Since $(*, \bar{1}, \bar{1}, 1)$ is $-\mathbf{e}_2$ boundary box, which means it moves to C_2 in $-\mathbf{e}2$ direction. Similar for $(\bar{1}, *, \bar{1}, 1)$. For the box indicated by $(*, 1, \bar{1}, 1)$ in C_1, one of its corners is $(-1, +1, -1, +1) \in B_0$. This point is identified with $(+1, -1, +1, -1) \in B_0$, which is a corner of $(1, *, 1, \bar{1})$ in C_2. The follow-ups are the same.

The results for all these processes are summarized in the tables below. The base corner is the corner in B_0 that demonstrates the adjacency. childId(B) and childId(B') are child indicators of the adjacent two boxes. The "relation" indicates the direction and its reverse between the two boxes.

Base Corner	childId(B)	relation	childId(B')
$(-1, -1, -1, -1)$	$(*, \bar{1}, \bar{1}, \bar{1})$	$\underset{-\mathbf{e}_1}{\overset{-\mathbf{e}_2}{\rightleftharpoons}}$	$(\bar{1}, *, \bar{1}, \bar{1})$
$(-1, +1, -1, -1)$	$(*, 1, \bar{1}, \bar{1})$	$\underset{+\mathbf{e}_1}{\overset{+\mathbf{e}_2}{\rightleftharpoons}}$	$(1, *, 1, 1)$
$(-1, -1, +1, -1)$	$(*, \bar{1}, 1, \bar{1})$	$\underset{-\mathbf{e}_1}{\overset{-\mathbf{e}_2}{\rightleftharpoons}}$	$(\bar{1}, *, 1, \bar{1})$
$(-1, +1, +1, -1)$	$(*, 1, 1, \bar{1})$	$\underset{+\mathbf{e}_1}{\overset{+\mathbf{e}_2}{\rightleftharpoons}}$	$(1, *, \bar{1}, 1)$
$(-1, -1, -1, +1)$	$(*, \bar{1}, \bar{1}, 1)$	$\underset{-\mathbf{e}_1}{\overset{-\mathbf{e}_2}{\rightleftharpoons}}$	$(\bar{1}, *, \bar{1}, 1)$
$(-1, +1, -1, +1)$	$(*, 1, \bar{1}, 1)$	$\underset{+\mathbf{e}_1}{\overset{+\mathbf{e}_2}{\rightleftharpoons}}$	$(1, *, 1, \bar{1})$
$(-1, -1, +1, +1)$	$(*, \bar{1}, 1, 1)$	$\underset{-\mathbf{e}_1}{\overset{-\mathbf{e}_2}{\rightleftharpoons}}$	$(\bar{1}, *, 1, \bar{1})$
$(-1, +1, +1, +1)$	$(*, 1, 1, 1)$	$\underset{+\mathbf{e}_1}{\overset{+\mathbf{e}_2}{\rightleftharpoons}}$	$(1, *, \bar{1}, \bar{1})$

Table E.1: Table between C_1 and C_2 (C_w and C_x)

Base Corner	childId(B)	relation	childId(B')
$(-1,-1,-1,-1)$	$(*,\bar{1},\bar{1},\bar{1})$	$\overset{-\mathbf{e}_3}{\underset{-\mathbf{e}_1}{\rightleftharpoons}}$	$(\bar{1},\bar{1},*,\bar{1})$
$(-1,+1,-1,-1)$	$(*,1,\bar{1},\bar{1})$	$\overset{-\mathbf{e}_3}{\underset{-\mathbf{e}_1}{\rightleftharpoons}}$	$(\bar{1},1,*,\bar{1})$
$(-1,-1,+1,-1)$	$(*,\bar{1},1,\bar{1})$	$\overset{+\mathbf{e}_3}{\underset{+\mathbf{e}_1}{\rightleftharpoons}}$	$(1,1,*,1)$
$(-1,+1,+1,-1)$	$(*,1,1,\bar{1})$	$\overset{+\mathbf{e}_3}{\underset{+\mathbf{e}_1}{\rightleftharpoons}}$	$(\bar{1},1,*,\bar{1})$
$(-1,-1,-1,+1)$	$(*,\bar{1},\bar{1},1)$	$\overset{-\mathbf{e}_3}{\underset{-\mathbf{e}_1}{\rightleftharpoons}}$	$(\bar{1},\bar{1},*,1)$
$(-1,+1,-1,+1)$	$(*,1,\bar{1},1)$	$\overset{-\mathbf{e}_3}{\underset{-\mathbf{e}_1}{\rightleftharpoons}}$	$(\bar{1},1,*,1)$
$(-1,-1,+1,+1)$	$(*,\bar{1},1,1)$	$\overset{+\mathbf{e}_3}{\underset{+\mathbf{e}_1}{\rightleftharpoons}}$	$(1,1,*,\bar{1})$
$(-1,+1,+1,+1)$	$(*,1,1,1)$	$\overset{+\mathbf{e}_3}{\underset{+\mathbf{e}_1}{\rightleftharpoons}}$	$(1,\bar{1},*,\bar{1})$

Table E.2: Table between C_1 and C_3 (C_w and C_y)

Base Corner	childId(B)	relation	childId(B')
$(-1,-1,-1,-1)$	$(*,\bar{1},\bar{1},\bar{1})$	$\overset{-\mathbf{e}_4}{\underset{-\mathbf{e}_1}{\rightleftharpoons}}$	$(\bar{1},\bar{1},\bar{1},*)$
$(-1,+1,-1,-1)$	$(*,1,\bar{1},\bar{1})$	$\overset{-\mathbf{e}_4}{\underset{-\mathbf{e}_1}{\rightleftharpoons}}$	$(\bar{1},1,\bar{1},*)$
$(-1,-1,+1,-1)$	$(*,\bar{1},1,\bar{1})$	$\overset{-\mathbf{e}_4}{\underset{-\mathbf{e}_1}{\rightleftharpoons}}$	$(\bar{1},\bar{1},1,*)$
$(-1,+1,+1,-1)$	$(*,1,1,\bar{1})$	$\overset{-\mathbf{e}_4}{\underset{-\mathbf{e}_1}{\rightleftharpoons}}$	$(\bar{1},1,1,*)$
$(-1,-1,-1,+1)$	$(*,\bar{1},\bar{1},1)$	$\overset{+\mathbf{e}_4}{\underset{+\mathbf{e}_1}{\rightleftharpoons}}$	$(1,1,1,*)$
$(-1,+1,-1,+1)$	$(*,1,\bar{1},1)$	$\overset{+\mathbf{e}_4}{\underset{+\mathbf{e}_1}{\rightleftharpoons}}$	$(1,\bar{1},1,*)$
$(-1,-1,+1,+1)$	$(*,\bar{1},1,1)$	$\overset{+\mathbf{e}_4}{\underset{+\mathbf{e}_1}{\rightleftharpoons}}$	$(1,1,\bar{1},*)$
$(-1,+1,+1,+1)$	$(*,1,1,1)$	$\overset{+\mathbf{e}_4}{\underset{+\mathbf{e}_1}{\rightleftharpoons}}$	$(1,\bar{1},\bar{1},*)$

Table E.3: Table between C_1 and C_4 (C_w and C_z)

Base Corner	childId(B)	relation	childId(B')
$(-1,-1,-1,-1)$	$(\overline{1},*,\overline{1},\overline{1})$	$\underset{-\mathbf{e}_2}{\overset{-\mathbf{e}_3}{\rightleftharpoons}}$	$(\overline{1},\overline{1},*,\overline{1})$
$(+1,-1,-1,-1)$	$(1,*,\overline{1},\overline{1})$	$\underset{-\mathbf{e}_2}{\overset{-\mathbf{e}_3}{\rightleftharpoons}}$	$(1,\overline{1},*,\overline{1})$
$(-1,-1,+1,-1)$	$(\overline{1},*,1,\overline{1})$	$\underset{+\mathbf{e}_2}{\overset{+\mathbf{e}_3}{\rightleftharpoons}}$	$(1,1,*,1)$
$(+1,-1,+1,-1)$	$(1,*,1,\overline{1})$	$\underset{+\mathbf{e}_2}{\overset{+\mathbf{e}_3}{\rightleftharpoons}}$	$(\overline{1},1,*,1)$
$(-1,-1,-1,+1)$	$(\overline{1},*,\overline{1},1)$	$\underset{-\mathbf{e}_2}{\overset{-\mathbf{e}_3}{\rightleftharpoons}}$	$(\overline{1},\overline{1},*,1)$
$(+1,-1,-1,+1)$	$(1,*,\overline{1},1)$	$\underset{-\mathbf{e}_2}{\overset{-\mathbf{e}_3}{\rightleftharpoons}}$	$(1,\overline{1},*,1)$
$(-1,-1,+1,+1)$	$(\overline{1},*,1,1)$	$\underset{+\mathbf{e}_2}{\overset{+\mathbf{e}_3}{\rightleftharpoons}}$	$(1,1,*,\overline{1})$
$(+1,-1,+1,+1)$	$(1,*,1,1)$	$\underset{+\mathbf{e}_2}{\overset{+\mathbf{e}_3}{\rightleftharpoons}}$	$(\overline{1},1,*,\overline{1})$

Table E.4: Table between C_2 and C_3 (C_x and C_y)

Base Corner	childId(B)	relation	childId(B')
$(-1,-1,-1,-1)$	$(\overline{1},*,\overline{1},\overline{1})$	$\underset{-\mathbf{e}_2}{\overset{-\mathbf{e}_4}{\rightleftharpoons}}$	$(\overline{1},\overline{1},\overline{1},*)$
$(+1,-1,-1,-1)$	$(1,*,\overline{1},\overline{1})$	$\underset{-\mathbf{e}_2}{\overset{-\mathbf{e}_4}{\rightleftharpoons}}$	$(1,\overline{1},\overline{1},*)$
$(-1,-1,+1,-1)$	$(\overline{1},*,1,\overline{1})$	$\underset{-\mathbf{e}_2}{\overset{-\mathbf{e}_4}{\rightleftharpoons}}$	$(\overline{1},\overline{1},1,*)$
$(+1,-1,+1,-1)$	$(1,*,1,\overline{1})$	$\underset{-\mathbf{e}_2}{\overset{-\mathbf{e}_4}{\rightleftharpoons}}$	$(1,\overline{1},1,*)$
$(-1,-1,-1,+1)$	$(\overline{1},*,\overline{1},1)$	$\underset{+\mathbf{e}_2}{\overset{+\mathbf{e}_4}{\rightleftharpoons}}$	$(1,1,1,*)$
$(+1,-1,-1,+1)$	$(1,*,\overline{1},1)$	$\underset{+\mathbf{e}_2}{\overset{+\mathbf{e}_4}{\rightleftharpoons}}$	$(\overline{1},1,1,*)$
$(-1,-1,+1,+1)$	$(\overline{1},*,1,1)$	$\underset{+\mathbf{e}_2}{\overset{+\mathbf{e}_4}{\rightleftharpoons}}$	$(1,1,\overline{1},*)$
$(+1,-1,+1,+1)$	$(1,*,1,1)$	$\underset{+\mathbf{e}_2}{\overset{+\mathbf{e}_4}{\rightleftharpoons}}$	$(\overline{1},1,\overline{1},*)$

Table E.5: Table between C_2 and C_4 (C_x and C_z)

Base Corner	childId(B)	relation	childId(B')
$(-1,-1,-1,-1)$	$(\bar{1},\bar{1},*,\bar{1})$	$\overset{-\mathbf{e}_4}{\underset{-\mathbf{e}_3}{\rightleftharpoons}}$	$(\bar{1},\bar{1},\bar{1},*)$
$(+1,-1,-1,-1)$	$(1,\bar{1},*,\bar{1})$	$\overset{-\mathbf{e}_4}{\underset{-\mathbf{e}_3}{\rightleftharpoons}}$	$(1,\bar{1},\bar{1},*)$
$(-1,+1,-1,-1)$	$(\bar{1},1,*,\bar{1})$	$\overset{-\mathbf{e}_4}{\underset{-\mathbf{e}_3}{\rightleftharpoons}}$	$(\bar{1},1,\bar{1},*)$
$(+1,+1,-1,-1)$	$(1,1,*,\bar{1})$	$\overset{-\mathbf{e}_4}{\underset{-\mathbf{e}_3}{\rightleftharpoons}}$	$(1,1,\bar{1},*)$
$(-1,-1,-1,+1)$	$(\bar{1},\bar{1},*,1)$	$\overset{+\mathbf{e}_4}{\underset{+\mathbf{e}_3}{\rightleftharpoons}}$	$(1,1,1,*)$
$(+1,-1,-1,+1)$	$(1,\bar{1},*,1)$	$\overset{+\mathbf{e}_4}{\underset{+\mathbf{e}_3}{\rightleftharpoons}}$	$(\bar{1},1,1,*)$
$(-1,+1,-1,+1)$	$(\bar{1},1,*,1)$	$\overset{+\mathbf{e}_4}{\underset{+\mathbf{e}_3}{\rightleftharpoons}}$	$(1,\bar{1},1,*)$
$(+1,+1,-1,+1)$	$(1,1,*,1)$	$\overset{+\mathbf{e}_4}{\underset{+\mathbf{e}_3}{\rightleftharpoons}}$	$(\bar{1},\bar{1},1,*)$

Table E.6: Table between C_3 and C_4 (C_y and C_z)

E.3 Tree-Path Indicator

Example E.3.1. We begin with $\mathcal{T}(B_0^t)$.

Suppose we have a subdivision tree for an interval I_0, denoted by $\mathcal{T}(I_0)$. The child indicator for each interval $I \in \mathcal{T}(I_0)$ is then only one number in $\{1,\bar{1}\}$. We define a **tree-path** of I a string β of $\{1,\bar{1}\}$, such that the i-th bit in the string is the child indicator of I', which is the ancestor of I such that $\text{depth}(I') = i$. The tree path for the root is the empty string ϵ. For each box $B^t \in \mathcal{T}(B_0^t)$, a **tree-path indicator** is a vector of strings, denoted by $\text{pathID}(B^t)$, such that the $j-th$ component is the tree-path for $\text{Proj}_{\widehat{j}}(B^t)$, where $\text{Proj}_{\widehat{j}}(B^t)$ is the interval representing the j-th component of B^t. For example, suppose we have a leaf given by 3 total splits where under each level, the ancestor is $(\bar{1},1,1)$, $(1,1,\bar{1})$, $(\bar{1},\bar{1},\bar{1})$. Then the tree-path indicator is

$$\text{pathID}(B^t) = (\bar{1}1\bar{1}, 11\bar{1}, 1\bar{1}\bar{1}).$$

As a result, the depth of a box under total splits the length of the string of each tree-path. The length of ϵ is 0. For simplicity, we call tree-path "path" for the rest of the example.

For each path string, we call the string where every bit is $\bar{1}$ the **negative boundary** (e.g. "$\overline{11111111}$") and the string where every bit is 1 the **positive boundary** (e.g. "111111111"). As a result, a box B is a **d**-boundary of B_0^t, if and only if for each component j, if $d_j = 1$, then its j-th path string is a positive boundary, if $d_j = -1$, then its j-th path string is a negative boundary. The **bar** of a string β is flipping all alphabets in the string (e.g. if $\mathrm{bar}(1\bar{1}11\bar{1}) = \bar{1}1\overline{11}1$). The **flip** of a string is flipping its last alphabet (e.g. if $\mathrm{flip}(1\bar{1}11\bar{1}) = 1\bar{1}111$). The **adj** of a string is flipping consecutive alphabets from the end of the string such that they are all $\bar{1}$ or all 1 until reach the beginning of the string or a different alphabet in the string (e.g. $\mathrm{adj}(1\bar{1}111) = 11\overline{111}$).

Given a subdivision tree of intervals $\mathcal{T}(I_0)$ the $+1$-neighbor of I is the interval represented by $\mathrm{adj}(\mathrm{pathID}(I))$ if $\mathrm{pathID}(I)$ is ending with 1 and is the interval represented by $\mathrm{flip}(\mathrm{pathID}(I))$ if $\mathrm{pathID}(I)$ is ending with $\bar{1}$. The latter case represents a $+1$-sibling, while the former case represents that I is the $+1$ boundary of the interval represented by the string before its consecutive 1s. If there is no such interval in $\mathcal{T}(I_0)$, then it is the interval represented by the longest substring reading from the left. For example, suppose that $\mathrm{pathID}(I) = \overline{11}1\bar{1}1111$, $\mathrm{adj}(\mathrm{pathID}(I)) = \overline{11}1\overline{11111}$. If there is no interval whose path string is $\overline{11}1\overline{11111}$, then it may be $\bar{1}$, or $\overline{11}$, or $\overline{11}1$, or $\overline{11}11$, ... The one existing with the longest such path string is the $+1$-neighbor. The case for -1-neighbor is symmetric.

For a subdivision tree of boxes $\mathcal{T}(B_0^t)$, the **d**-cousin is the box given by path indicator that is correspondingly adj or flip to each component according to the value of **d** at that component. For example, the path indicator for $\mathbf{e}_2$-cousin of $(\bar{1}1\bar{1}, 11\bar{1}, 1\overline{11})$ is $(\bar{1}1\bar{1}, 111, 1\overline{11})$, for $-\mathbf{e}_3$-cousin of $(\bar{1}1\bar{1}, 11\bar{1}, 1\overline{11})$ is $(\bar{1}1\bar{1}, 11\bar{1}, \bar{1}11)$. Note that to get the principal neighbor of B^t, one only needs to trace the string from the left until it is a leaf or the string ends, since its the **d**-neighbor B' with maximal depth subject to $\mathrm{depth}(B') \leq \mathrm{depth}(B)$.

Now we turn to the case $\mathcal{T}(\widehat{SO}(3))$. By Proposition 6.2.6, the child indicator for $+\mathbf{e}_j$-cousin to $+\mathbf{e}_j$ boundary is flipped in each level of each component. Then, to get the corresponding path indicator, one only needs to operate the strings by **bar**. For $-\mathbf{e}_j$-cousin, the child indicators are unchanged, so the path indicator is also unchanged. Using all these techniques, all operations for child indicators can be transferred to a corresponding operation for path indicators.

The power of a path indicator is even stronger than above. For each path string β, we have an integer representation $\mathrm{int}(\beta)$. It's done by the following:

Integer Representation for Tree-Path strings int(β)
Input: string β.
Output: $\mathrm{int}(\beta)$.
Substitute each $\bar{1}$ in β by '0' and form β_1.
Add a 1 before the string β_1 and form β_2.
Return the integer whose binary string is β_2.

For example,

$$\bar{1}1\bar{1} \to 010 \to (1010)_2 \to (10)_{10},$$

so $\mathrm{int}(\bar{1}1\bar{1}) = 10$.

The operators depth, bar, flip and adj can also be done by simple integer operations. When represented by integers, flip and adj can be uniformed into one operation. The integer of path string of +1-neighbor of I is just $\mathrm{int}(\mathrm{pathID}(I)) + 1$ and the integer of path string of -1-neighbor of I is just $\mathrm{int}(\mathrm{pathID}(I)) - 1$. We don't need to distinguish if the neighbor is a sibling or a cousin any more. The depth of an integer representation n is $\lfloor \log_2 n \rfloor$. The bar of an integer representation n is $2^{\mathrm{depth}(n)+1} - n$.

As a trick, to separate the child indicator at level k from a path string whose integer representation is n, one only need to return "$n >> (\mathrm{depth}(n) - k)\&1$" in C++, where 1 represents 1 and 0 represent $\bar{1}$.

Using integer representation, one can compress child indicators for 31 levels into a single integer typed variable, which saves the disk space a lot.

Appendix F

Experimental Results

In this chapter, we present data collected from the demo program developed in this work. We test three scenarios, including a totally blocked scene (BL), a two-turns scene (TT) and a guarded-hole scene (GH). In each scenario, we vary ε and apply different heuristics (Qtype), and collect data including the time costs (in seconds), outputs including a Path (P) or NO-PATH (N) or "Memory Limit Exceeded" (M), the number of `FREE` (F), `STUCK` (S), `MIXED` (M) and ε-small (ε) boxes, the number of boxes that were split (#Box), and the minimums width of the boxes in the subdivision trees (Min w). To avoid disk overload ("Memory Limit Exceeded"), we set the maximum boxes that can be split to be 170001. Time values beyond this number of splits are denoted as $> 30000(s)$.

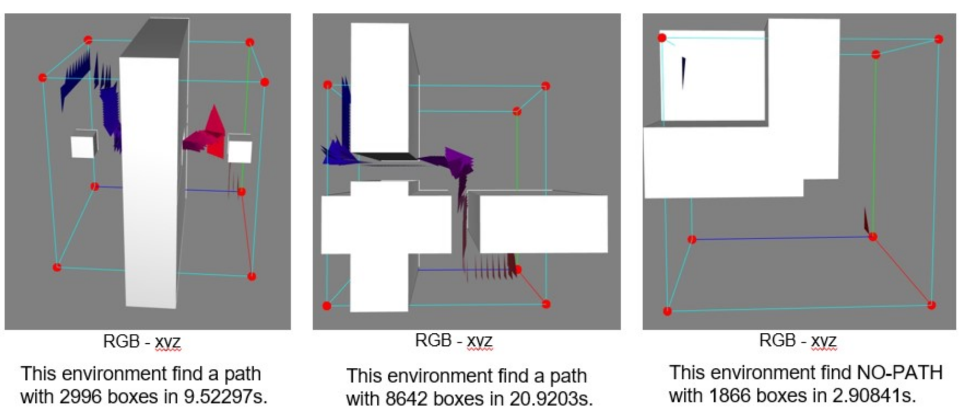


Figure F.1: Demo Performances of the "GuardedHole", "TwoTurn" and "Blocked" senarios.

This page shows the data of the "Blocked" scene (three walls blocking α and β).

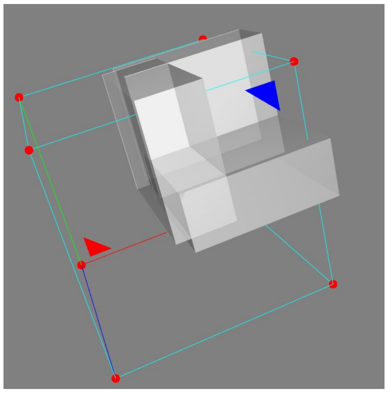


Figure F.2: Front view of the Blocked (BL) scene.

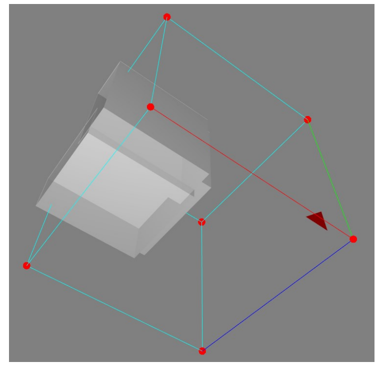


Figure F.3: Back view of the Blocked (BL) scene.

Env	Start Conf.	Goal Conf.	ε	Qtype	time(s)	Path	F/S/M/ε	#Box	Min w
BL	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.1	recursive	2.2352	N	338/4/3431/0	431	0.125
BL	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.2	recursive	0.5789	N	338/4/879/2552	431	0.125
BL	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.1	random	> 30000	M	84115/31930/385114/962827	170001	0.0625
BL	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.2	random	10.3094	N	2427/1153/10668/59552	9225	0.125

Table F.1: Data of the "Blocked" scene.

This page shows the data of the "TwoTurn" scene (two walls with gaps towards different directions).

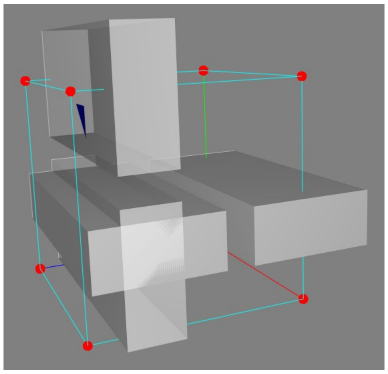


Figure F.4: Front view of the TwoTurn (TT) scene.

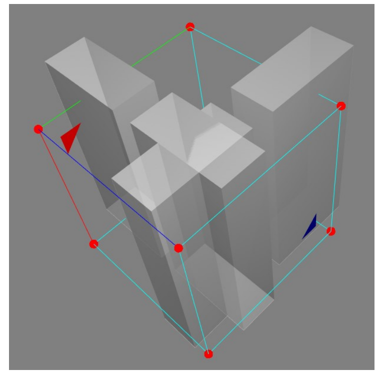


Figure F.5: Back view of the TwoTurn (TT) scene.

Env	Start Conf.	Goal Conf.	ε	Qtype	time(s)	Path	F/S/M/ε	#Box w	Min w
TT	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.1	recursive	20.9203	Y	11837/4495/32046/24086	8642	0.0625
TT	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.2	recursive	0.0484	N	8/0/64/8	10	0.125
TT	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.4	recursive	0.0442	N	0/0/56/16	9	0.25
TT	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.1	random	> 30000	M	85153/24773/473496/763368	170001	0.0625
TT	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.2	random	0.0482	N	8/0/64/8	10	0.125
TT	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.4	random	0.0395	N	0/0/56/16	9	0.25

Table F.2: Data of the "TwoTurn" scene.

This page shows the data of the “GuardedHole” scene (a hole on a wall with two boxes guarding the hole).

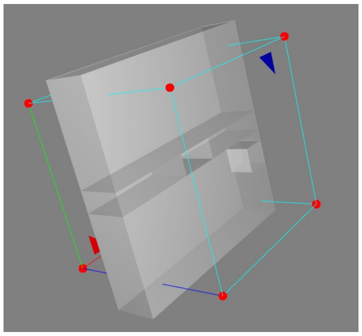


Figure F.6: Front view of the GuardedHole (GH) scene.

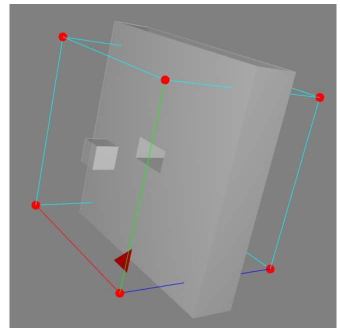


Figure F.7: Back view of the GuardedHole (GH) scene.

Env	Start Conf.	Goal Conf.	ε	Qtype	time(s)	Path	F/S/M/ε	#Box w	Min w
GH	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.1	recursive	9.5230	Y	3115/1069/8621/6728	2996	0.0625
GH	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.2	recursive	34.5132	N	7714/0/9112/26880	4793	0.125
GH	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.4	recursive	0.9968	N	321/0/800/2368	398	0.25
GH	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.1	random	> 30000	M	80238/42383/397983/833168	170001	0.0625
GH	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.2	random	52.5353	N	3036/1825/15307/86120	13286	0.125
GH	$\begin{pmatrix} -3.50 \\ -3.50 \\ -3.50 \\ 1.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$	$\begin{pmatrix} 3.50 \\ 3.50 \\ 3.50 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{pmatrix}$	0.4	random	6.2883	N	660/296/2972/16744	2584	0.25

Table F.3: Data of the “GuardedHole” scene.

As a result, the recursive heuristic method gives a much better performance than the purely random heuristic method. However, our goal is to give out a real-time path planning method. There is still future work to do to accelerate the recursive method.

Bibliography

[1] Jr. Alexander Kirillov. *An Introduction to Lie Groups and Lie Algebras.* Cambridge studies in advanced mathematics, 2008. ISBN: 978-1-316-61410-5.

[2] M.F. Atiyah and I.G. MacDonald. *Introduction To Commutative Algebra.* Addison-Wesley series in mathematics. Avalon Publishing, 1994.

[3] F. Avnaim, J.D. Boissonnat, and B. Faverjon. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 1656–1661 vol.3, 1988.

[4] A.T. Fomenko B.A. Dubrovin and Sergei Petrovich Novikov. *Modern Geometry – Methods and Applications.* Original Russian edition published by Nauka, 1979. Part I. The Geometry of Surfaces, Transformation Groups, and Fields. Second Edition.

[5] Julien Basch, Leonidas J Guibas, David Hsu, and An Thai Nguyen. Disconnection proofs for motion planning. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 2, pages 1765–1772. IEEE, 2001.

[6] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in real algebraic geometry.* Springer, 2006.

[7] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry.* SPRINGER, Berlin, Heidelberg, 2006.

[8] Huck Bennett and Chee Yap. Amortized analysis of smooth quadtrees in all dimensions. *Computational Geometry*, 63:20–39, 2017.

[9] G. Besson, J. Lohkamp, P. Pansu, and P. Petersen. *Riemannian geometry.* Springer, 2006.

[10] Temistocle Birsan and Dan Tiba. One hundred years since the introduction of the set distance by dimitrie pompeiu. In *System Modeling and Optimization: Proceedings of the 22nd IFIP TC7 Conference held from July 18–22, 2005, in Turin, Italy 22*, pages 35–39. Springer, 2006.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, U.K., 2004.

[12] R. A. Brooks and T. Lozano-Perez. A subdivision algorithm in configuration space for findpath with rotation. *Systems Man and Cybernetics IEEE Transactions on*, SMC-15(2):224–233, 1982.

[13] Rodney A Brooks and Tomas Lozano-Perez. A subdivision algorithm in configuration space for findpath with rotation. *IEEE Transactions on Systems, Man, and Cybernetics*, (2):224–233, 1985.

[14] John Canny. *The complexity of robot motion planning.* MIT press, 1988.

[15] H. Choset, K. M. Lynch, S. Hutchinson, G.Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations.* MIT Press, Boston, 2005.

[16] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–271, 1959.

[17] M.S.E. Din and E. Schost. A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets. *J.ACM*, 63(6):48:1–48:7, 1983.

[18] Ioannis Z Emiris and Menelaos I Karavelas. The predicates of the apollonius diagram: algorithmic analysis and implementation. *Computational Geometry*, 33(1-2):18–57, 2006.

[19] H. Eves. *An Introduction to the History of Mathematics.* The Saunders series. Saunders College Pub., 1990.

[20] J. A. Fridy. *Introductory Analysis - The Theory of Calculus.* Academic Press, 2000. ISBN 9780122676550.

[21] Andrew S Glassner. Space subdivision for fast ray tracing. *IEEE Computer Graphics and applications*, 4(10):15–24, 1984.

[22] Dan Halperin, Oren Salzman, and Micha Sharir. Algorithmic motion planning. In *Handbook of Discrete and Computational Geometry*, pages 1311–1342. Chapman and Hall/CRC, 2017.

[23] Ching-Hsiang Hsu, Yi-Jen Chiang, and Chee Yap. Rods and rings: Soft subdivision planner for $\mathbf{R}\hat{}3$ x $\mathbf{S}\hat{}2$. In *Proc. 35th (SoCG 2019)*, pages 43:1–43:17, 2019.

[24] Ching Hsiang Hsu, John Paul Ryan, and Chee Yap. Path planning for simple robots using soft subdivision search. In Sandor Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry, SoCG 2016*, Leibniz International Proceedings in Informatics, LIPIcs, pages 68.1–68.5. Schloss Dagstuhl- Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, June 2016.

[25] D. Hsu, J.-C. Latombe, and R. Motwani. *Path planning in expansive configuration spaces*, volume 3. 1997.

[26] Du Q. Huynh. Metrics for 3d rotations: Comparison and analysis. *J Math Imaging Vis*, 35:155–164, 2009.

[27] T. Jacob, Schwartz, Micha, and Sharir. On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds. *Advances in Appl. Math.*, 4:298–351, 1983.

[28] Lydia E Kavraki, Jean-Claude Latombe, Rajeev Motwani, and Prabhakar Raghavan. Randomized query processing in robot path planning. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 353–362, 1995.

[29] Vladlen Koltun. Pianos are not flat: Rigid motion planning in three dimensions. In *Symposium on Discrete Algorithms: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, volume 23, pages 505–514. Citeseer, 2005.

[30] S. M. LaValle. *Planning Algorithms.* Cambridge University Press, Cambridge, U.K., 2006.

[31] Carl W. Lee and Wendy Weber. Recovering lexicographic triangulations, 2017.

[32] Ji Yeong Lee and Howie Choset. Sensor-based planning for a rod-shaped robot in three dimensions: Piecewise retracts of r3× s2. *The International Journal of Robotics Research*, 24(5):343–383, 2005.

[33] Sihui Li and Neil T Dantam. Exponential convergence of infeasibility proofs for kinematic motion planning. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 294–311. Springer, 2022.

[34] Sihui Li and Neil T Dantam. Scaling infeasibility proofs via concurrent, codimension-one, locally-updated coxeter triangulation. *IEEE Robotics and Automation Letters*, 2023.

[35] Ming C Lin and Dinesh Manocha. Collision detection. In *Handbook of Data Structures and Applications*, pages 889–902. Chapman and Hall/CRC, 2018.

[36] Tomás Lozano-Pérez and Michael A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, (11):560–570, 1979.

[37] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control.* Cambridge University Press, 2017. ISBN: 9781107156302.

[38] Kurt Mehlhorn. From theory to Library of Efficient Data types and Algorithms (LEDA) and algorithm engineering. In Cristian Calude, editor, *The Human Face of Computing*, pages 59–72. Imperical College Press, 2015. Chapter in Volume.

[39] John M.Lee. *Introduction to Smooth Manifolds. Second Edition.* Graduate Texts in Mathematics, 2013. http://www.springer.com/series/136.

[40] Ramon E Moore. *Interval analysis*, volume 4. prentice-Hall Englewood Cliffs, 1966.

[41] Ramon E Moore, R Baker Kearfott, and Michael J Cloud. *Introduction to interval analysis*. SIAM, 2009.

[42] James Munkres. *Elements of Algebraic Topology*. Addison-Wesley, 1984. ISBN 0-201-04586-9.

[43] Michal Nowakiewicz. Mst-based method for 6dof rigid body motion planning in narrow passages. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5380–5385. IEEE, 2010.

[44] Colm Ó'Dúnlaing, Micha Sharir, and Chee K Yap. Retraction: A new approach to motion-planning. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 207–220, 1983.

[45] Colm Ó'Dunlaing, Micha Sharir, and Chee K Yap. Generalized voronoi diagrams for moving a ladder. i: Topological analysis. *Communications on Pure and Applied Mathematics*, 39(4):423–483, 1986.

[46] Colm Ó'Dunlaing, Micha Sharir, and Chee K Yap. Generalized voronoi diagrams for moving a ladder ii: efficient computation of the diagram. *Algorithmica*, 2:27–59, 1987.

[47] Colm Ó'Dúnlaing and Chee K Yap. A "retraction" method for planning the motion of a disc. *Journal of Algorithms*, 6(1):104–111, 1985.

[48] Andreas Orthey, Constantinos Chamzas, and Lydia E. Kavraki. Sampling-based motion planning: A comparative review. *Annual Reviews.*, 25:6–9, 2023.

[49] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[50] Mohab Safey el Din and Éric Schost. A baby steps/giant steps probabilistic algorithm for computing roadmaps in smooth bounded real hypersurface. *Discrete & Computational Geometry*, 45(1):181–220, 2011.

[51] Jacob T Schwartz and Micha Sharir. On the "piano movers'" problem i. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on pure and applied mathematics*, 36(3):345–398, 1983.

[52] Jacob T Schwartz and Micha Sharir. On the "piano movers" problem. ii. general techniques for computing topological properties of real algebraic manifolds. *Advances in applied Mathematics*, 4(3):298–351, 1983.

[53] Jacob T Schwartz and Micha Sharir. On the piano movers' problem: V. the case of a rod moving in three-dimensional space amidst polyhedral obstacles. *Communications on Pure and Applied Mathematics*, 37(6):815–848, 1984.

[54] Vikram Sharma and Chee K. Yap. Robust geometric computation. In Jacob E. Goodman, Joseph O'Rourke, and Csaba Tóth, editors, *Handbook of Discrete and Computational Geometry*, chapter 45, pages 1189–1224. Chapman & Hall/CRC, Boca Raton, FL, 3rd edition, 2017.

[55] Nir Shvalb, Boaz Ben Moshe, and Oded Medina. A real-time motion planning algorithm for a hyper-redundant set of mechanisms. *Robotica*, 31(8):1327–1335, 2013.

[56] Cong Wang, Yi-Jen Chiang, and Chee Yap. On soft predicates in subdivision motion planning. *Comput. Geometry: Theory and Appl. (Special Issue for SoCG'13)*, 48(8):589–605, September 2015.

[57] A. Wittig. *An Introduction to Differential Algebra and the Differential Algebra Manifold Representation.* Astrodynamics Network AstroNet-II, 2016.

[58] Chee Yap. Soft subdivision search in motion planning. In A. Aladren et al., editor, *Proc. 1st Workshop on Robotics Challenge and Vision (RCV 2013)*, 2013. A Computing Community Con-

sortium (CCC) **B**est Paper Award, Robotics Science and Systems Conf. (RSS 2013), Berlin. In arXiv:1402.3213.

[59] Chee Yap. Soft subdivision search and motion planning, II: Axiomatics. In *Frontiers in Algorithmics*, volume 9130 of *Lecture Notes in Comp. Sci.*, pages 7–22. Springer, 2015. Plenary talk at 9th FAW. Guilin, China. Aug. 3-5, 2015.

[60] Chee-Keng Yap. Algorithmic motion planning. In *Advances in Robotics, Vol. 1: Algorithmic and geometric issues*, pages 95–143. Lawrence Erlbaum Associates, 1987.

[61] Jihun Yu, Chee Yap, Zilin Du, Sylvain Pion, and Hervé Brönnimann. The design of core 2: A library for exact numeric computation in geometry and algebra. In Komei Fukuda, Joris van der Hoeven, Michael Joswig, and Nobuki Takayama, editors, *Mathematical Software – ICMS 2010*, pages 121–141, Berlin, Heidelberg, 2010. Springer.

[62] W. Zeng and R. L. Church. Finding shortest paths on real road networks: the case for A*. *International Journal of Geographical Information Science*, 23(4):531–543, 2009.

[63] Liangjun Zhang, Young J Kim, and Dinesh Manocha. Efficient cell labelling and path non-existence computation using c-obstacle query. *The International Journal of Robotics Research*, 27(11-12):1246–1257, 2008.

[64] Xinya Zhang, Robert Belfer, Paul G Kry, and Etienne Vouga. C-space tunnel discovery for puzzle path planning. *ACM Transactions on Graphics (TOG)*, 39(4):104–1, 2020.

[65] Zhaoqi Zhang, Yi-Jen Chiang, and Chee Yap. Theory and explicit design of a path planner for an SE(3) robot. *WAFR*, 41(2), 2024.

[66] Bo Zhou, Yi-Jen Chiang, and Chee Yap. Soft subdivision motion planning for complex planar robots. *Computational Geometry*, 92, January 2021. Article 101683. Originally, in 26th ESA, 2018.

[67] David Zhu and J-C Latombe. Constraint reformulation in a hierarchical path planner. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 1918–1923. IEEE, 1990.

[68] David Zhu, Jean-Claude Latombe, et al. *New heuristic algorithms for efficient hierarchical path planning*. Department of Computer Science, Stanford University, 1989.